



Benutzerhandbuch

# Amazon CodeCatalyst



# Amazon CodeCatalyst: Benutzerhandbuch

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

Was ist Amazon CodeCatalyst? .....	1
Was kann ich damit machen? CodeCatalyst .....	1
Wie fange ich an mit CodeCatalyst? .....	2
Erfahren Sie mehr über CodeCatalyst .....	2
Konzepte .....	3
AWS Builder-ID-Leerzeichen in CodeCatalyst .....	4
Bereiche, die den Identitätsverbund unterstützen in CodeCatalyst .....	4
Projekte .....	4
Blueprints .....	4
Kontoverbindungen .....	5
VPC-Verbindungen .....	5
AWS Builder-ID .....	5
Benutzerprofile in CodeCatalyst .....	6
Quell-Repositoryys .....	6
Commits .....	7
Entwicklungsumgebungen .....	7
Workflows .....	8
Aktionen .....	8
Problembereiche .....	8
Persönliche Zugriffstoken (PATs) .....	9
Rollen .....	9
Einrichtung .....	10
Registrieren und Erstellen von Ressourcen .....	12
Erstellen Ihres ersten Bereichs und Ihrer ersten IAM-Rollen .....	13
Nehmen Sie eine Einladung an .....	19
Eine Einladung annehmen und eine Builder-ID erstellen AWS .....	20
Melden Sie sich mit Ihrer AWS Builder-ID an .....	22
Vertrauenswürdige Geräte .....	22
Nehmen Sie eine E-Mail-Einladung zur Anmeldung mit SSO an .....	22
Mit SSO anmelden .....	23
Alle Bereiche und Projekte für einen Benutzer anzeigen .....	23
CodeCatalyst Profile anzeigen und verwalten .....	24
Ihr CodeCatalyst Profil ansehen .....	25
Das CodeCatalyst Profil eines anderen Benutzers anzeigen .....	25

Aktualisierung Ihres Profils .....	26
Dein CodeCatalyst Passwort ändern .....	27
Einrichtung für die AWS CLI Verwendung von CodeCatalyst .....	28
Tutorials für die ersten Schritte .....	30
Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen .....	31
Voraussetzungen .....	33
Schritt 1: Erstellen Sie das moderne dreistufige Webanwendungsprojekt .....	34
Schritt 2: Laden Sie jemanden zu Ihrem Projekt ein .....	36
Schritt 3: Erstelle Aufgaben, an denen du gemeinsam arbeiten kannst, und verfolge die Arbeit .....	36
Schritt 4: Sehen Sie sich Ihr Quell-Repository an .....	37
Schritt 5: Erstellen Sie eine Entwicklungsumgebung mit einem Testzweig und nehmen Sie eine schnelle Codeänderung vor .....	38
Schritt 6: Sehen Sie sich den Workflow an, der die moderne Anwendung erstellt .....	40
Schritt 7: Bitten Sie andere, Ihre Änderungen zu überprüfen .....	43
Schritt 8: Schließen Sie das Problem .....	47
Bereinigen von -Ressourcen .....	48
Referenz .....	49
Tutorial: Mit einem leeren Projekt beginnen .....	51
Voraussetzungen .....	51
Erstellen Sie ein leeres Projekt .....	51
Erstellen Sie ein Quell-Repository .....	52
Erstellen Sie einen Workflow zum Erstellen, Testen und Bereitstellen einer Codeänderung ...	53
Laden Sie jemanden zu Ihrem Projekt ein .....	54
Erstellen Sie Probleme, an denen Sie gemeinsam arbeiten können, und verfolgen Sie die Arbeit .....	54
Tutorial: Verwenden von Funktionen für generative KI .....	55
Voraussetzungen .....	56
Hinzufügen einer automatisch generierten Zusammenfassung beim Erstellen einer Pull- Anforderung .....	57
Erstellen einer Zusammenfassung der Kommentare zu Codeänderungen in einer Pull- Anforderung .....	60
Erstellen eines Problems und Zuweisen zu Amazon Q .....	61
Bereinigen von -Ressourcen .....	68
Tutorial: Erstellen einer Full-Stack-Anwendung mit zusammensetzbaren PDK-Blueprints .....	68



Voraussetzungen .....	70
Schritt 1: Erstellen Sie ein Monorepo-Projekt .....	71
Schritt 2: Fügen Sie dem Projekt die Type Safe-API hinzu .....	72
Schritt 3: Fügen Sie eine Cloudscape React-Website für das Projekt hinzu .....	74
Schritt 4: Generieren Sie die Infrastruktur für die Bereitstellung der Anwendung in der AWS-Cloud .....	75
Schritt 5: Richten Sie einen DevOps Workflow für die Bereitstellung Ihres Projekts ein .....	77
Schritt 6: Bestätigen Sie den Release-Workflow und sehen Sie sich Ihre Website an .....	79
Arbeiten Sie zusammen und iterieren Sie am PDK-Projekt .....	85
Leerzeichen .....	102
Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt .....	104
Einen Bereich bearbeiten .....	107
Löschen eines Leerzeichens .....	108
Überwachung der Aktivität für einen Bereich .....	109
Verwaltung AWS-Konten für einen Raum .....	109
Ein AWS-Konto zu einem Leerzeichen hinzufügen .....	110
Hinzufügen von IAM-Rollen zu Kontoverbindungen .....	114
Hinzufügen der Kontoverbindung und der IAM-Rollen zu Ihrer Bereitstellungsumgebung .....	116
Kontoverbindungen anzeigen .....	117
Konto aus einem Bereich entfernen (in CodeCatalyst) .....	118
Verwaltung von IAM-Rollen für verbundene Konten .....	119
CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> Rolle .....	120
AWSRoleForCodeCatalystSupport Rolle .....	121
Eine IAM-Rolle erstellen und die CodeCatalyst Vertrauensrichtlinie verwenden .....	122
Verwaltung von Space-Benutzern .....	124
Mitglieder in einem Bereich anzeigen .....	124
Einen Benutzer direkt in einen Bereich einladen .....	126
Eine Einladung für einen Raum stornieren .....	127
Die Rolle eines Space-Mitglieds ändern .....	128
Ein Space-Mitglied entfernen .....	128
Entfernen oder Ändern der Rolle für einen Benutzer mit der Space-Administratorrolle .....	129
Teams verwalten .....	131
Ein Team erstellen .....	131
Ein Team ansehen .....	134
Verwaltung der Space-Rolle für ein Team .....	134
Eine Projektrolle für ein Team verwalten .....	135

Direktes Hinzufügen eines Benutzers zu einem Team .....	136
Direktes Entfernen eines Benutzers aus einem Team .....	137
Einem Team eine SSO-Gruppe hinzufügen .....	137
Ein Team löschen .....	138
Verwaltung von Maschinenressourcen .....	138
Maschinenressourcen anzeigen .....	139
Deaktivieren Sie die Maschinenressourcen .....	139
Aktivieren Sie Maschinenressourcen .....	141
Verwaltung von Entwicklungsumgebungen für einen Bereich .....	141
Entwicklungsumgebungen für deinen Bereich anzeigen .....	142
Eine Entwicklungsumgebung für deinen Bereich bearbeiten .....	143
Eine Entwicklungsumgebung für deinen Bereich beenden .....	144
Löschen einer Entwicklungsumgebung für deinen Bereich .....	144
Kontingente für Leerzeichen .....	145
Projekte .....	147
Erstellen eines Projekts .....	148
Ein Projekt mit einem Blueprint erstellen .....	149
Ein leeres Projekt erstellen .....	150
Ein Projekt mit einem verknüpften GitHub Repository erstellen .....	150
Ein Projekt ansehen .....	153
Projektaufgaben und Entwicklungsumgebungen anzeigen .....	154
Alle Projekte ansehen .....	155
.....	155
Projekteinstellungen anzeigen .....	155
Zu einem anderen Projekt wechseln in CodeCatalyst .....	156
Löschen eines Projekts .....	156
Verwaltung von Projektmitgliedern .....	156
Mitglieder in einem Projekt anzeigen .....	157
Einen Benutzer zu Ihrem Projekt einladen .....	158
Eine Einladung stornieren .....	160
Einen Benutzer aus Ihrem Projekt entfernen .....	160
Eine Einladung für ein Projekt annehmen oder ablehnen .....	161
Teams für Projekte leiten .....	161
Ein Team zu einem Projekt hinzufügen .....	162
Eine Projektrolle für ein Team verwalten .....	162
Eine Projektrolle für ein Team entfernen .....	163

Verwaltung von Maschinenressourcen .....	163
Maschinenressourcen anzeigen .....	164
Deaktivieren Sie die Maschinenressourcen .....	164
Aktivieren Sie Maschinenressourcen .....	166
Kontingente für Projekte .....	166
Working with notifications .....	167
Wie funktionieren Benachrichtigungen? .....	168
Erste Schritte mit Slack-Benachrichtigungen .....	170
Benachrichtigungen verwalten .....	173
Blueprints .....	179
Ein Projekt mit einem Blueprint erstellen .....	180
Blueprints in einem Projekt anwenden und deren Zuordnung aufheben .....	180
Einen Blueprint auf ein Projekt anwenden .....	181
Aufheben der Zuordnung eines Blueprints zu einem Projekt .....	182
Aktualisierung eines Blueprints in einem Projekt .....	182
Bearbeiten einer Beschreibung für einen Blueprint in einem Projekt .....	184
Als Blueprint-Benutzer mit Lifecycle Management arbeiten .....	185
Verwenden des Lebenszyklusmanagements für bestehende Projekte .....	185
Verwenden des Lebenszyklusmanagements für mehrere Blueprints in einem Projekt .....	185
Umgang mit Konflikten in Lifecycle-Pull-Requests .....	186
Abmeldung von Änderungen im Lebenszyklusmanagement .....	186
Überschreiben des Lebenszyklusmanagements eines Blueprints in einem Projekt .....	186
Referenz zum Projekt-Blueprint .....	187
Verfügbare Blueprints .....	187
Informationen zum Projekt-Blueprint finden .....	192
Arbeiten mit benutzerdefinierten Blueprints .....	192
Konzepte für benutzerdefinierte Vorlagen .....	193
Erste Schritte mit benutzerdefinierten Blueprints .....	197
Tutorial: Eine React-Anwendung erstellen und aktualisieren .....	202
Arbeiten mit Lebenszyklusmanagement als Blueprint-Autor .....	210
Entwickeln eines benutzerdefinierten Blueprints .....	216
Veröffentlichen eines benutzerdefinierten Blueprints .....	248
Details, Versionen und Projekte eines benutzerdefinierten Blueprints anzeigen .....	253
Hinzufügen und Entfernen eines benutzerdefinierten Blueprints in einem Bereich .....	255
Veröffentlichungsberechtigungen für einen benutzerdefinierten Blueprint verwalten .....	256
Versionen für einen benutzerdefinierten Blueprint verwalten .....	257

---

Löschen eines veröffentlichten benutzerdefinierten Blueprints oder einer veröffentlichten Version .....	258
Arbeiten mit Abhängigkeiten und Tools .....	259
Beitragend .....	262
Kontingente für Entwürfe .....	262
Quell-Repositoryn .....	263
Konzepte für Quell-Repositoryn .....	265
Projekte .....	4
Quell-Repositorys .....	265
Entwicklungsumgebungen .....	7
Persönliche Zugriffstoken (PATs) .....	9
Branches .....	267
Standardzweige .....	267
Commits .....	7
Pull-Anforderungen .....	268
Überarbeitungen .....	268
Workflows .....	8
Einrichtung .....	269
Installieren Sie Git .....	270
Erstellen Sie ein persönliches Zugriffstoken .....	270
Erste Schritte mit Quell-Repositorys .....	271
Ein Projekt mithilfe eines Blueprints erstellen .....	271
Die Repositorys für ein Projekt anzeigen .....	273
Erstellen einer Entwicklungsumgebung .....	274
Erstellen einer Pull-Anforderung .....	276
Eine Pull-Anfrage zusammenführen .....	279
Den bereitgestellten Code anzeigen .....	280
Bereinigen von Ressourcen .....	280
Arbeiten mit Quell-Repositorys .....	281
Erstellen eines Quell-Repositorys .....	282
Ein Quell-Repository verlinken .....	283
Quell-Repository anzeigen .....	284
Einstellungen für ein Quell-Repository bearbeiten .....	286
Klonen eines Quell-Repositorys .....	287
Löschen eines Quell-Repositorys .....	289
Arbeiten mit Zweigen .....	290

Einen Zweig erstellen und löschen .....	291
Den Standardzweig für ein Repository anzeigen und ändern .....	293
Filiialregeln verwalten .....	294
Git-Befehle für Branches .....	297
Filialen und Details anzeigen .....	299
Mit Dateien arbeiten .....	300
Eine Datei erstellen oder hinzufügen .....	300
Eine Datei ansehen .....	303
Bearbeiten einer Datei .....	304
Eine Datei umbenennen oder löschen .....	305
Verwenden von Pull-Anforderungen .....	305
Erstellen einer Pull-Anforderung .....	307
Pull-Requests anzeigen .....	312
Genehmigungsregeln verwalten .....	314
Einen Pull-Request überprüfen .....	316
Aktualisierung einer Pull-Anfrage .....	319
Eine Pull-Anfrage zusammenführen .....	321
Einen Pull-Request schließen .....	325
Mit Commits arbeiten .....	326
Commits für einen Branch anzeigen .....	327
Ändern der Anzeige von Commits (CodeCatalystKonsole) .....	328
Kontingente für Quell-Repositorys .....	328
Entwicklungsumgebungen .....	334
Erstellen einer Entwicklungsumgebung .....	335
Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen .....	336
Erstellen einer Entwicklungsumgebung in CodeCatalyst .....	336
Eine Entwicklungsumgebung in einer IDE erstellen .....	339
Anhalten einer Entwicklungsumgebung .....	340
Fortsetzen einer Entwicklungsumgebung .....	341
Eine Entwicklungsumgebung bearbeiten .....	343
Löschen einer Entwicklungsumgebung .....	344
Verbindung zu einer Entwicklungsumgebung über SSH herstellen .....	345
Konfiguration Ihrer Entwicklungsumgebung .....	347
Bearbeiten einer Repository-Dev-Datei für eine Entwicklungsumgebung in CodeCatalyst ....	349
Bearbeiten einer Repository-Dev-Datei für eine Entwicklungsumgebung in einer IDE .....	350
Eine Repository-Dev-Datei für eine Entwicklungsumgebung verschieben .....	350

Wiederherstellungsmodus .....	350
Devfile-Funktionen werden unterstützt von CodeCatalyst .....	351
Beispiel: Konfiguration der Dev-Datei für Ihre Entwicklungsumgebung .....	351
Devfile-Befehle .....	352
Devfile-Ereignisse .....	353
Devfile-Komponenten .....	354
Universelle Devfile-Images .....	355
Verwenden von Entwicklungsumgebungen mit einer VPC-Verbindung .....	360
Verwenden von Entwicklungsumgebungen mit einer IDE .....	361
Kontingente für Entwicklungsumgebungen .....	362
Pakete .....	363
Konzepte für Pakete .....	364
Pakete .....	364
Paket-Namespaces .....	364
Versionen von Paketen .....	364
Objekte .....	365
Paket-Repositorys .....	365
Gateway-Repositorys .....	365
Upstream-Repositoryn .....	366
Arbeiten mit Paket-Repositorys .....	366
Ein Paket-Repository erstellen .....	366
Verbindung zu einem Paket-Repository herstellen .....	367
Bearbeiten eines Paket-Repositorys .....	367
Löschen eines Paket-Repositorys .....	368
Arbeiten mit Upstream-Repositorys .....	368
Hinzufügen eines Upstream-Repositorys .....	369
Die Suchreihenfolge von Upstream-Repositorys bearbeiten .....	370
Eine Paketversion mit Upstream-Repositorys anfordern .....	371
Entfernen eines Upstream-Repositorys .....	374
Verbindung zu öffentlichen externen Repositorien herstellen .....	375
Unterstützte externe Paket-Repositorys und ihre Gateway-Repositorys .....	376
Mit Paketen arbeiten .....	376
Veröffentlichen von Paketen .....	377
Details zur Paketversion anzeigen .....	378
Löschen einer Paketversion .....	378
Den Status einer Paketversion aktualisieren .....	379

Steuerelemente für den Paketursprung bearbeiten .....	380
Verwenden von npm .....	386
Konfiguration und Verwendung von npm .....	387
Handhabung von NPM-Tags .....	397
Kontingente für Pakete .....	398
Erstellen, Testen und Bereitstellen mit Workflows .....	399
Informationen zur Workflow-Definitionsdatei .....	399
Verwenden der visuellen Editoren und der YAML-Editoren der CodeCatalyst Konsole .....	402
Workflows entdecken .....	403
Details zur Workflow-Ausführung anzeigen .....	404
Nächste Schritte .....	405
Konzepte für Workflows .....	405
Workflows .....	405
Workflow-Definitionsdateien .....	406
Aktionen .....	406
Aktionsgruppen .....	406
-Artefakte .....	407
Datenverarbeitung .....	407
Umgebungen .....	407
Berichte .....	407
Ausführungen .....	408
Quellen .....	408
Variablen .....	408
Workflow-Auslöser .....	408
Erste Schritte mit Workflows .....	409
Voraussetzungen .....	409
Schritt 1: Erstellen und konfigurieren Sie Ihren Workflow .....	410
Schritt 2: Speichern Sie Ihren Workflow mit einem Commit .....	412
Schritt 3: Ausführungsergebnisse anzeigen .....	412
(Optional) Schritt 4: Bereinigen .....	413
Codequalität und Bereitstellungsstatus nach Commit anzeigen .....	413
Bauen mithilfe von Workflows .....	415
Wie erstelle ich eine Anwendung? .....	415
Vorteile der Build-Aktion .....	416
Alternativen zur Build-Aktion .....	417
Die Build-Aktion wird hinzugefügt .....	417

Anzeigen der Ergebnisse .....	419
Tutorial: Artefakte auf Amazon S3 hochladen .....	419
Testen mithilfe von Workflows .....	429
Typen von Testberichten .....	429
Die Testaktion wird hinzugefügt .....	432
Berichte konfigurieren .....	433
Benutzen universal-test-runner .....	449
Bewährte Methoden .....	451
Mit Tests arbeiten .....	455
Bereitstellung mithilfe von Workflows .....	457
Wie stelle ich eine Anwendung bereit? .....	458
Liste der Bereitstellungsaktionen .....	459
Vorteile von Bereitstellungsaktionen .....	460
Alternativen zur Bereitstellung von Aktionen .....	460
Tutorial: Bereitstellen mit CloudFormation .....	461
Tutorial: Auf Amazon ECS bereitstellen .....	489
Tutorial: Auf Amazon EKS bereitstellen .....	525
Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen .....	559
Aktion „In Amazon ECS bereitstellen“ hinzufügen .....	562
Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen .....	565
Die Aktion „AWS CDK Deploy“ wird hinzugefügt .....	569
Arbeiten mit Bereitstellungen .....	576
Mit Workflows arbeiten .....	588
Einen Workflow erstellen, bearbeiten und löschen .....	589
Anzeigen des Workflow-Status .....	594
Verwenden von Aktionen .....	596
Mit Artefakten arbeiten .....	660
Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen .....	674
Arbeiten mit Umgebungen .....	699
Arbeiten mit Datei-Caching .....	707
Mit Paketen arbeiten .....	712
Mit Läufen arbeiten .....	717
Arbeiten mit Secrets .....	731
Mit Quellen arbeiten .....	735
Arbeiten mit Auslösern .....	739
Arbeiten mit Variablen .....	757



Referenz zur Workflow-Definition .....	784
Beispiel für eine Workflow-Definitionsdatei .....	785
Richtlinien und Konventionen zur Syntax .....	786
Immobilien auf höchstem Niveau .....	789
Aktionsreferenz zum Erstellen und Testen .....	801
Referenz zur Aktion „Amazon S3 veröffentlichen“ .....	832
Referenz zur Aktion „AWS CDK bootstrap“ .....	842
Aktionsreferenz „AWS CDK bereitstellen“ .....	854
„AWS Lambda Aufruf“-Aktionsreferenz .....	871
Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“ .....	886
Referenz zur Aktion „Deploy to Amazon ECS“ .....	905
Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“ .....	918
Referenz zur Aktion „GitHub Aktionen“ .....	928
Referenz zur Aktion „Aufgabendefinition rendern“ .....	949
Kontingente für Workflows .....	957
Problembereiche .....	960
Probleme, Konzepte .....	961
Aktive Probleme .....	962
Archivierte Probleme .....	962
Beauftragter .....	962
Benutzerdefinierte Felder .....	962
Schätzung .....	962
Problem .....	963
Label (Bezeichnung) .....	963
Priorität .....	963
Status und Statuskategorien .....	963
Aufgaben .....	964
Ansichten .....	964
Ein Problem wird erstellt .....	964
Bewährte Methoden für Probleme, die Amazon Q zugewiesen wurden .....	967
Probleme bearbeiten und gemeinsam bearbeiten .....	969
Ein Problem bearbeiten .....	969
Arbeiten mit Anhängen .....	971
Aufgaben zu bestimmten Themen verwalten .....	973
Ein Problem als blockiert oder entsperrt markieren .....	973
Kommentare hinzufügen, bearbeiten oder löschen .....	974

Ein Problem wird bearbeitet .....	976
Ein Problem archivieren .....	978
Probleme finden und anzeigen .....	979
Sie suchen nach einem Problem .....	980
Probleme beim Sortieren .....	980
Probleme gruppieren .....	981
Probleme beim Filtern .....	982
Eine Problemansicht erstellen .....	983
Probleme beim Exportieren .....	983
Einstellungen für Probleme konfigurieren .....	984
Aktivieren oder Deaktivieren mehrerer Beauftragter .....	984
Konfiguration der Problemaufwandsschätzung .....	985
Statuses .....	985
Labels .....	987
Benutzerdefinierte Felder .....	989
Anlagen anzeigen und verwalten .....	989
Kontingente für Probleme .....	990
Identität, Berechtigungen und Zugriff .....	992
Arbeiten mit -Rollen .....	993
Rollentypen .....	993
Für jede Rolle verfügbare Berechtigungen .....	996
Benutzerrollen anzeigen und ändern .....	1043
Verwaltung persönlicher Zugriffstoken .....	1045
PATs erstellen .....	1045
PATs anzeigen .....	1048
PATs löschen .....	1049
Multi-Faktor-Authentifizierung (MFA) bei Amazon CodeCatalyst .....	1050
Wie registriere ich ein Gerät für die Verwendung mit Multi-Faktor-Authentifizierung .....	1051
Authentifizierungsanwendungen .....	1053
Ihre MFA-Geräte ändern .....	1054
Sicherheit .....	1055
Datenschutz .....	1056
CodeCatalyst und Identity and Access Management .....	1059
Compliance-Validierung .....	1126
Ausfallsicherheit .....	1127
Sicherheit der Infrastruktur .....	1127

Konfigurations- und Schwachstellenanalyse .....	1128
Ihre Daten und Ihr Datenschutz bei Amazon CodeCatalyst .....	1128
Bewährte Methoden für Workflow-Aktionen .....	1129
CodeCatalyst Vertrauensmodell .....	1130
Überwachung bei Amazon CodeCatalyst .....	1132
CodeCatalyst API-Aufrufe an verbundene Konten protokollieren in AWS CloudTrail .....	1135
Zugriff auf protokollierte Ereignisse in CodeCatalyst .....	1143
Kontingente für Identität, Erlaubnis und Zugriff .....	1146
Fehlerbehebung .....	1148
Probleme bei der Anmeldung .....	1148
Probleme bei der Anmeldung .....	1149
Probleme beim Abmelden .....	1149
Ich erhalte die Fehlermeldung „Rolle existiert nicht“ für einen fehlgeschlagenen Workflow .	1150
Ich erhalte einen Rollenfehler für einen fehlgeschlagenen Workflow .....	1150
Ich muss die IAM-Rolle in einem Projekt-Workflow aktualisieren .....	1151
Wie fülle ich ein Support-Formular aus? .....	1151
Erweiterungen .....	1152
Konzepte für Erweiterungen .....	1152
Erweiterungen .....	1152
CodeCatalyst katalog .....	1152
Erweiterungen installieren und deinstallieren .....	1152
Eine Erweiterung installieren .....	1153
Deinstallation einer Erweiterung .....	1154
Verwenden von GitHub Repositorien in CodeCatalyst .....	1154
Schnellstart: GitHub Repositorys verwenden in CodeCatalyst .....	1156
GitHub Konten verwalten .....	1159
Verwaltung von Repositorien GitHub .....	1162
Verknüpfte Repositorys GitHub anzeigen .....	1167
Verwendung verknüpfter GitHub Repositorys in Workflows .....	1168
Verwendung von IP-Adressbeschränkungen in GitHub Enterprise Cloud .....	1170
Blockieren von GitHub Pull-Request-Zusammenführungen, wenn Workflows fehlschlagen	1171
Verwendung von Jira-Problemen in CodeCatalyst .....	1171
Schnellstart: Jira-Probleme verwenden in CodeCatalyst .....	1172
Verwaltung von Jira-Seiten .....	1176
Verwaltung von Jira-Projekten .....	1179
Jira-Issues mit Pull-Requests verknüpfen .....	1182

CodeCatalyst Ereignisse in Jira anzeigen .....	1183
Jira-Probleme suchen in CodeCatalyst .....	1184
Suche .....	1185
Verfeinern Sie Ihre Suchabfrage .....	1186
Verfeinerung nach Typ .....	1186
Verfeinerung nach Feldern .....	1187
Verfeinerung mit booleschen Operatoren .....	1187
Verfeinerung nach Projekten .....	1188
Überlegungen bei der Arbeit mit der Suche .....	1188
Referenz zu durchsuchbaren Feldern .....	1189
Fehlerbehebung .....	1195
Behebung allgemeiner Zugriffsprobleme .....	1195
Ich habe mein Passwort vergessen .....	1195
Amazon CodeCatalyst ist teilweise oder vollständig nicht verfügbar .....	1196
Ich kann kein Projekt erstellen in CodeCatalyst .....	1196
Behebung von Support-Problemen .....	1196
Ich erhalte eine Fehlermeldung, wenn ich auf Amazon AWS Support zugreife CodeCatalyst .....	1196
Ich kann keine Anfragen für technischen Support für meine Unterkunft erstellen .....	1197
Mein Konto für Supportanfragen ist nicht mehr mit meinem Bereich in verbunden CodeCatalyst .....	1197
Ich kann keinen Support-Fall für ein anderes AWS-Service in AWS Support for Amazon eröffnen CodeCatalyst .....	1198
Amazon CodeCatalyst ist teilweise oder vollständig nicht verfügbar .....	1196
Ich kann kein Projekt erstellen in CodeCatalyst .....	1196
Ich möchte Feedback einreichen in CodeCatalyst .....	1196
Behebung von Quell-Repositorys .....	1199
Ich habe die maximale Speicherkapazität für meinen Speicherplatz erreicht und erhalte Warnungen oder Fehler .....	1200
Ich erhalte eine Fehlermeldung, wenn ich versuche, ein CodeCatalyst Amazon-Quell- Repository zu klonen oder zu pushen .....	1200
Ich erhalte eine Fehlermeldung, wenn ich versuche, einen Commit oder einen Push in ein CodeCatalyst Amazon-Quell-Repository durchzuführen .....	1201
Ich benötige ein Quell-Repository für mein Projekt .....	1202
Mein Quell-Repository ist brandneu, enthält aber einen Commit .....	1202
Ich möchte einen anderen Zweig als meinen Standardzweig .....	1202

Ich erhalte E-Mails über Aktivitäten in Pull-Requests .....	1203
Ich habe mein persönliches Zugriffstoken (PAT) vergessen .....	1203
Eine Pull-Anfrage zeigt nicht die Änderungen an, die ich erwarte .....	1203
Ein Pull-Request zeigt den Status Nicht zusammenführbar .....	1204
Problembeseitigung bei Projekten und Blueprints .....	1205
Java-API mit AWS Fargate Blueprint, fehlende Abhängigkeiten für Apache-Maven-3.8.6 ...	1205
Moderner dreistufiger Blueprint-Workflow für Webanwendungen OnPullRequests schlägt mit einem Berechtigungsfehler für Amazon fehl CodeGuru .....	1206
Sie suchen immer noch nach einer Lösung für Ihr Problem? .....	1210
Behebung von Arbeitsabläufen .....	1210
Wie behebe ich die Meldung „Der Workflow ist inaktiv“? .....	1211
Wie behebe ich den Fehler „Workflow-Definition enthält <i>keine</i> Fehler“? .....	1212
Wie behebe ich die Fehler „Anmeldeinformationen konnten nicht gefunden werden“ und ExpiredToken „“? .....	1214
Wie behebe ich die Fehler „Es konnte keine Verbindung zum Server hergestellt werden“? .....	1216
Warum fehlen CodeDeploy Felder im Visual Editor? .....	1216
Wie behebe ich Fehler bei den IAM-Fähigkeiten? .....	1217
Wie behebe ich „npm install“ -Fehler? .....	1219
Warum haben mehrere Workflows denselben Namen? .....	1223
Kann ich meine Workflow-Definitionsdateien in einem anderen Ordner speichern? .....	1223
Wie füge ich meinem Workflow nacheinander Aktionen hinzu? .....	1224
Warum wird mein Workflow erfolgreich validiert, schlägt aber zur Laufzeit fehl? .....	1224
Die automatische Erkennung erkennt keine Berichte für meine Aktion .....	1224
Meine Aktion schlägt bei automatisch erkannten Berichten fehl, nachdem ich Erfolgskriterien konfiguriert habe .....	1225
Autodiscovery generiert Berichte, die ich nicht haben möchte .....	1226
Auto-Discovery generiert viele kleine Berichte für ein einzelnes Test-Framework .....	1226
Die unter CI/CD aufgeführten Workflows stimmen nicht mit denen im Quell-Repository überein .....	1226
Ich kann keine Workflows erstellen oder aktualisieren .....	1228
Problembeseitigung bei der Suche .....	1228
Ich kann in meinem Projekt keinen Benutzer finden .....	1228
Ich sehe in meinem Projekt oder Bereich nicht, wonach ich suche .....	1229
Die Anzahl der Suchergebnisse ändert sich ständig, wenn ich durch die Seiten navigiere ..	1229
Meine Suchabfrage wird nicht abgeschlossen .....	1229
Fehlerbeseitigung bei zugeordneten Konten .....	1230

Meine AWS-Konto Verbindungsanforderung erhält einen ungültigen Token-Fehler .....	1230
Mein Amazon- CodeCatalyst Projektworkflow schlägt mit einem Fehler für das konfigurierte Konto, die Umgebung oder die IAM-Rolle fehl .....	1231
Ich benötige ein zugeordnetes Konto, eine Rolle und eine Umgebung, um ein Projekt zu erstellen .....	1232
Ich kann nicht auf die Seite Amazon CodeCatalyst Spaces in der zugreifen AWS Management Console .....	1233
Ich möchte ein anderes Konto als mein Abrechnungskonto .....	1202
Fehlerbehebung bei Entwicklungsumgebungen .....	1234
Die Erstellung meiner Entwicklungsumgebung war aufgrund eines Problems mit den Kontingenten nicht erfolgreich .....	1235
Ich kann keine Änderungen aus meiner Entwicklungsumgebung in einen bestimmten Zweig in einem Repository übertragen .....	1235
Meine Entwicklungsumgebung wurde nicht wieder aufgenommen .....	1236
Die Verbindung zu meiner Entwicklungsumgebung wurde unterbrochen .....	1236
Meine mit dem VPC verbundene Entwicklungsumgebung ist ausgefallen .....	1236
Ich kann nicht finden, in welchem Verzeichnis sich mein Projekt befindet .....	1237
Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen .....	1237
Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen, da meine lokale SSH-Konfiguration fehlt .....	1237
Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen, da ich Probleme mit meinem Profil AWS Config habe codecatalyst .....	1238
Fehlerbehebung bei IDEs .....	1238
Problembehandlung bei Devfiles .....	1240
Beheben von -Problemen .....	1242
Ich kann keinen Beauftragten für mein Problem auswählen .....	1242
Fehlerbehebung AWS CLI und SDK-Probleme .....	1243
Ich erhalte eine Fehlermeldung, wenn ich über eine Befehlszeile oder ein Terminal aws codecatalyst eingebe, dass es sich um eine ungültige Auswahl handelt .....	1243
Ich erhalte einen Anmeldeinformationsfehler, wenn ich aws codecatalyst Befehle ausführe .....	1243
CodeCatalyst Gesundheitsbericht .....	1245
CodeCatalyst Konzepte für Gesundheitsberichte .....	1245
Vorfall .....	1246
Status .....	1246
Betroffene Funktionen .....	1246
Aktualisiert am .....	1246

---

AWS Support für Amazon CodeCatalyst .....	1247
Abrechnung AWS Support für Amazon CodeCatalyst .....	1247
Deinen Bereich AWS Support für Amazon einrichten CodeCatalyst .....	1250
Zugriff auf Support für CodeCatalyst in AWS Management Console .....	1251
Einen CodeCatalyst Supportfall erstellen in CodeCatalyst .....	1252
Lösung eines Support-Falls in CodeCatalyst .....	1255
Wiedereröffnung eines Support-Falls in CodeCatalyst .....	1256
Kontingente .....	1258
Dokumentverlauf .....	1260
AWS Glossar .....	1285
.....	mcclxxxvi

# Was ist Amazon CodeCatalyst?

Amazon CodeCatalyst ist ein integrierter Service für Softwareentwicklungsteams, die kontinuierliche Integrations- und Bereitstellungspraktiken in ihren Softwareentwicklungsprozess integrieren. CodeCatalyst vereint die Tools, die Sie benötigen, an einem Ort. Mit Tools für kontinuierliche Integration/Continuous Delivery (CI/CD) können Sie Arbeit planen, gemeinsam am Code arbeiten und Anwendungen erstellen, testen und bereitstellen. Sie können Ressourcen auch in Ihre Projekte integrieren, indem Sie Ihre AWS Ressourcen mit Ihrem AWS-Konten Bereich verbinden. CodeCatalyst Indem Sie alle Phasen und Aspekte Ihres Anwendungslebenszyklus in einem Tool verwalten, können Sie Software schnell und zuverlässig bereitstellen.

In erstellen Sie einen Bereich CodeCatalyst, der Ihr Unternehmen, Ihre Abteilung oder Gruppe repräsentiert, und dann erstellen Sie Projekte, die die Ressourcen enthalten, die Sie zur Unterstützung Ihrer Entwicklungsteams und Aufgaben benötigen. CodeCatalystRessourcen sind innerhalb von Projekten strukturiert, die sich innerhalb von Räumen befinden. Damit Teams schnell loslegen können, CodeCatalyst bietet es Sprach- oder Tool-basierte Projekt-Blueprints. Wenn Sie ein Projekt anhand eines Projekt-Blueprints erstellen, enthält das Projekt Ressourcen wie ein Quell-Repository mit Beispielcode, Build-Skripts, Bereitstellungsaktionen, virtuelle Server oder serverlose Ressourcen und mehr.

## Was kann ich damit machen? CodeCatalyst

Sie und Ihr Entwicklungsteam können CodeCatalyst damit jeden Aspekt der Softwareentwicklung durchführen, von der Planung Ihrer Arbeit bis hin zur Bereitstellung Ihrer Anwendungen. Sie können CodeCatalyst für Folgendes verwenden:

- Iterieren und gemeinsam am Code arbeiten — Arbeiten Sie gemeinsam mit Ihrem Team an Code mit Verzweigungen, Zusammenführungen, Pull-Requests und Kommentaren in Ihren Quellcode-Repositorys. Erstellen Sie Entwicklungsumgebungen, um schnell an Code zu arbeiten, ohne dass Sie Repositorys klonen oder Verbindungen zu Repositorys einrichten müssen.
- Erstellen, testen und implementieren Sie Ihre Anwendung mit Workflows — Konfigurieren Sie Workflows mit Aktionen zum Erstellen, Testen und Bereitstellen, um die kontinuierliche Integration und Bereitstellung Ihrer Anwendungen zu gewährleisten. Sie können Workflows entweder manuell starten oder sie so konfigurieren, dass sie auf der Grundlage von Ereignissen wie Code-Pushs oder dem Erstellen oder Schließen von Pull-Requests automatisch gestartet werden.



- Priorisiere die Arbeit deines Teams mit der Problemverfolgung — Verwende Probleme, um Backlogs zu erstellen und den Status laufender Aufgaben mit Boards zu überwachen. Ein wichtiger Bestandteil der Softwareentwicklung ist es, einen gesunden Rückstand an Aufgaben zu erstellen und aufrechtzuerhalten, an denen Ihr Team arbeiten kann.
- Richten Sie Überwachung und Benachrichtigungen ein — Überwachen Sie die Teamaktivität und den Ressourcenstatus und konfigurieren Sie Benachrichtigungen, um über wichtige Änderungen auf dem Laufenden zu bleiben.

## Wie fange ich an mit CodeCatalyst?

Wenn Sie keinen Bereich haben oder erfahren möchten, wie Sie einen Bereich einrichten und verwalten, empfehlen wir Ihnen, mit dem [Amazon CodeCatalyst Administrator Guide zu beginnen](#).

Wenn Sie noch nicht mit der Arbeit in einem Projekt oder Bereich vertraut sind, empfehlen wir Ihnen, wie folgt zu beginnen:

- Überprüfung der [CodeCatalyst Konzepte](#)
- [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#)
- Erstellen Sie Ihr erstes Projekt, indem Sie die Schritte unter befolgen [Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen](#)

## Erfahren Sie mehr über CodeCatalyst

CodeCatalyst In diesem Benutzerhandbuch sowie in den folgenden Ressourcen erfahren Sie mehr über die Funktionen von:

- [AWS DevOps Blogartikel über Amazon CodeCatalyst](#)
- [Das Amazon CodeCatalyst API-Referenzhandbuch](#)
- [Das Amazon CodeCatalyst Action Development Kit Entwicklerhandbuch](#)
- [CodeCatalyst Häufig gestellte Fragen](#)
- [Zeugnisse](#)

# CodeCatalyst Konzepte

Machen Sie sich mit den wichtigsten Konzepten vertraut, um Ihre Zusammenarbeit und Anwendungsentwicklung bei Amazon zu beschleunigen CodeCatalyst. Zu diesen Konzepten gehören Begriffe, die in den Bereichen Quellcodeverwaltung, Continuous Integration und Continuous Delivery (CI/CD) sowie Modellierung und Konfiguration automatisierter Release-Prozesse verwendet werden.

Weitere konzeptionelle Informationen finden Sie in den folgenden Themen:

- [Konzepte für Quell-Repositoryen](#)
- [Workflow-Konzepte](#)

## Themen

- [AWS Builder-ID-Leerzeichen in CodeCatalyst](#)
- [Bereiche, die den Identitätsverbund unterstützen in CodeCatalyst](#)
- [Projekte](#)
- [Blueprints](#)
- [Kontoverbindungen](#)
- [VPC-Verbindungen](#)
- [AWS Builder-ID](#)
- [Benutzerprofile in CodeCatalyst](#)
- [Quell-Repositorys](#)
- [Commits](#)
- [Entwicklungsumgebungen](#)
- [Workflows](#)
- [Aktionen](#)
- [Problembereiche](#)
- [Persönliche Zugriffstoken \(PATs\)](#)
- [Rollen](#)

# AWS Builder-ID-Leerzeichen in CodeCatalyst

Der Space-Administrator lädt Benutzer dazu ein, CodeCatalyst indem er individuelle Einladungs-E-Mails von der Mitgliederseite aus sendet. Benutzer, die eingeladen sind oder sich registrieren, um ihre eigene AWS Builder-ID zu CodeCatalyst erstellen. Das Profil wird in AWS Builder ID verwaltet und als Benutzername und Profilvereinstellungen unter angezeigt CodeCatalyst.

## Bereiche, die den Identitätsverbund unterstützen in CodeCatalyst

Benutzer, die zu den SSO-Benutzern und -Gruppen für die IAM Identity Center-Instanz hinzugefügt wurden und im Identitätsspeicher verwaltet werden und über IAM Identity Center in Ihren Bereich eingeladen werden. Der Space-Administrator synchronisiert die CodeCatalyst Mitgliederseite für die neuesten Updates. Benutzer melden sich über das SSO-Anmeldeportal an, das in der IAM Identity Center-Instanz des Unternehmens eingerichtet ist. Bereiche, die den Identitätsverbund unterstützen, sind über die Identity Center-Anwendung und deren Zuordnung zur Identitätsspeicher-ID mit der Identitätsspeicher-Instanz verbunden.

## Projekte

Ein Projekt ist ein Gemeinschaftsprojekt CodeCatalyst , das Entwicklungsteams und Aufgaben unterstützt. Sobald Sie ein Projekt erstellt haben, können Sie Benutzer und Ressourcen hinzufügen, aktualisieren oder entfernen, Ihr Projekt-Dashboard anpassen und den Fortschritt der Arbeit Ihres Teams überwachen. Sie können mehrere Projekte in einem Bereich haben.

Weitere Informationen zu Projekten finden Sie unter [Projekte in CodeCatalyst](#).

## Blueprints

Ein Blueprint ist ein Projektsynthesizer, der Dateien und Abhängigkeiten zur Anwendungsunterstützung für Sie generiert und erweitert und Ihr CodeCatalyst Projekt in der Konsole erstellt. Sie wählen einen Projekttyp aus einer Auswahl von Blueprints in CodeCatalyst, sehen sich die README-Datei an und zeigen eine Vorschau des Projekt-Repositorys und der Ressourcen an, die generiert werden. Ihr Projekt wird anhand der im Blueprint angegebenen Basiskonfiguration generiert. Sie führen regelmäßig Synthesen zum Projekt-Blueprint durch, wodurch Ihre Projektdateien, z. B. Softwareabhängigkeiten, aktualisiert und Ressourcen neu generiert werden.

Projekte verwenden ein Tool namens Projen, um Projekte zu synthetisieren, indem die neuesten Projektaktualisierungen synchronisiert und Unterstützungsdateien generiert werden. Diese Dateien können `package.json`, `Makefile`, und mehr enthaltenes `lint`, je nach Anwendungstyp und Sprache. Mit Projekt-Blueprints können Dateien generiert werden, die AWS Ressourcen wie CDK-Konstrukte, AWS CloudFormation Vorlagen und Vorlagen unterstützen. AWS Serverless Application Model

Weitere Informationen zu Projekt-Blueprints finden Sie unter [Referenz zum Projekt-Blueprint](#)

## Kontoverbindungen

Eine Kontoverbindung verknüpft einen CodeCatalyst Bereich mit Ihrem AWS-Konto. Nachdem Ihre Kontoverbindung eingerichtet wurde, AWS-Konto wird der dem Bereich zur Verfügung gestellt. Sie können dann IAM-Rollen hinzufügen, CodeCatalyst damit es auf Ressourcen in Ihrem AWS-Konto zugreifen kann. Sie können diese Rollen auch für Ihre CodeCatalyst Workflow-Aktionen verwenden.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

## VPC-Verbindungen

Eine VPC-Verbindung ist eine CodeCatalyst Ressource, die alle Konfigurationen enthält, die Ihr Workflow für den Zugriff auf eine VPC benötigt. Space-Administratoren können im Namen von Space-Mitgliedern ihre eigenen VPC-Verbindungen in der CodeCatalyst Amazon-Konsole hinzufügen. Durch Hinzufügen einer VPC-Verbindung können Space-Mitglieder Workflow-Aktionen ausführen und Entwicklungsumgebungen erstellen, die den Netzwerkregeln entsprechen und auf Ressourcen in der zugehörigen VPC zugreifen können.

Weitere Informationen zu VPC-Verbindungen finden Sie unter [Managing Amazon Virtual Private Clouds](#) im CodeCatalyst Administratorhandbuch.

## AWS Builder-ID

Eine AWS Builder-ID ist eine persönliche Identität, mit der Sie sich registrieren und bei anderen teilnehmenden Anwendungen anmelden können. CodeCatalyst Es ist nicht dasselbe wie ein AWS-Konto. Ihre AWS Builder-ID verwaltet Metadaten wie Benutzeralias und E-Mail-Adresse. Ihre AWS Builder-ID ist eine eindeutige Identität, die Benutzer in allen Bereichen unterstützt CodeCatalyst.

Informationen zum Zugriff auf Ihr AWS Builder ID-Profil finden Sie unter [Aktualisierung Ihres Profils](#). Weitere Informationen zur AWS Builder ID finden Sie unter [AWS Builder ID](#) im Allgemeine AWS-Referenz.

Weitere Informationen zur Registrierung und Anmeldung finden Sie unter [Einrichten CodeCatalyst](#).

## Benutzerprofile in CodeCatalyst

Sie greifen auf Ihr CodeCatalyst Benutzerprofil zu, indem Sie die Profiloption aus der Dropdownliste unter Ihren Anmeldeinitialen auf einer beliebigen Seite in CodeCatalyst auswählen. Sie können persönliche Zugriffstoken (PATs) auf Ihrer Profilseite erstellen, aber Sie können PATs nur mit dem anzeigen oder löschen. AWS CLI Ihr Benutzername ist der Alias, den Sie bei der Registrierung gewählt haben. Sie können Ihren Benutzernamen nicht ändern. Um die Profilseite eines anderen CodeCatalyst Benutzers anzuzeigen, gehen Sie zum Tab Mitglieder für Ihr Projekt und wählen Sie den entsprechenden Benutzer aus.

Sie greifen auf Ihre AWS Builder-ID zu, indem Sie Ihr CodeCatalyst Profil aufrufen und dann zur AWS Builder-ID wechseln. Sie werden auf Ihre AWS Builder ID-Profilseite weitergeleitet. Der vollständige Name, die E-Mail-Adresse und das Passwort Ihres Profils werden über Ihre AWS Builder-ID verwaltet. Sie können diese Informationen auf der AWS Builder-ID-Seite bearbeiten. Sie haben diese Informationen bei der Registrierung eingegeben. Wenn Sie bereit sind, MFA für die Verwendung einer Authentifikatoranwendung für die Anmeldung einzurichten, verwenden Sie die AWS Builder-ID-Seite. Weitere Informationen zum Anzeigen Ihres AWS Builder ID-Profiles finden Sie unter [Aktualisierung Ihres Profils](#)

Weitere Informationen zur Registrierung und Anmeldung finden Sie unter [Einrichten CodeCatalyst](#).

## Quell-Repositorys

In einem Quell-Repository speichern Sie Code und Dateien für Ihr Projekt sicher. Es speichert auch den Versionsverlauf Ihrer Dateien. Standardmäßig wird ein Quell-Repository mit den anderen Benutzern in Ihrem CodeCatalyst Projekt geteilt. Sie können mehr als ein Quell-Repository für ein Projekt haben. Sie können Quell-Repositorys für Projekte erstellen oder ein vorhandenes Quell-Repository verknüpfen CodeCatalyst, das von einem anderen Dienst gehostet wird, sofern dieser Dienst von einer installierten Erweiterung unterstützt wird. Sie können beispielsweise ein GitHub Repository mit einem Projekt verknüpfen, nachdem Sie die GitHub Repository-Erweiterung installiert haben. Weitere Informationen finden Sie unter [Arbeiten mit Quell-Repositorys in CodeCatalyst](#) und [Schnellstart: GitHub Repositorys verwenden in CodeCatalyst](#).

In Quell-Repositorys werden auch Konfigurationsinformationen für Ihr CodeCatalyst Projekt gespeichert, z. B. die Konfigurationsdatei, die die Attribute und Aktionen Ihres CI/CD-Workflows definiert. Wenn Sie Ihr Projekt mithilfe eines Blueprints erstellen, wird ein Quell-Repository mit darin gespeicherten Projektkonfigurationsinformationen erstellt. Wenn Sie ein leeres Projekt erstellen, müssen Sie ein Quell-Repository erstellen, bevor Sie Ressourcen erstellen können, für die Konfigurationsinformationen erforderlich sind, z. B. Workflows.

Weitere Konzepte, die Ihnen bei der Arbeit mit Quell-Repositorys und der Quellcodeverwaltung helfen können, finden Sie unter [Konzepte für Quell-Repositoryn](#).

## Commits

Ein Commit ist eine Änderung an einer Datei oder einer Gruppe von Dateien. In der CodeCatalyst Amazon-Konsole speichert ein Commit Ihre Änderungen und überträgt sie in ein Quell-Repository. Der Commit enthält Informationen über die Änderung, einschließlich der Identität des Benutzers, der die Änderung vorgenommen hat, Uhrzeit und Datum der Änderung, den Commit-Titel und alle Nachrichten zu der Änderung. Weitere Informationen finden Sie unter [Mit Commits in Amazon arbeiten CodeCatalyst](#).

Im Kontext eines Quell-Repositorys in CodeCatalyst sind Commits Schnappschüsse der Änderungen am Inhalt Ihres Repositorys. Jedes Mal, wenn ein Benutzer eine Änderung festschreibt und diese pusht, werden Informationen CodeCatalyst gespeichert, darunter, wer die Änderung vorgenommen hat, Datum und Uhrzeit des Commits sowie die Änderungen, die im Rahmen des Commits vorgenommen wurden. Du kannst Commits auch Git-Tags hinzufügen, um bestimmte Commits leichter identifizieren zu können.

Weitere Informationen zu Commits finden Sie unter [Mit Commits in Amazon arbeiten CodeCatalyst](#)

## Entwicklungsumgebungen

Eine Entwicklungsumgebung ist eine cloudbasierte Entwicklungsumgebung, mit der Sie schnell CodeCatalyst an dem Code arbeiten können, der in den Quell-Repositorys Ihres Projekts gespeichert ist. Die in Ihrer Entwicklungsumgebung enthaltenen Projekttools und Anwendungsbibliotheken werden durch eine Dev-Datei im Quell-Repository Ihres Projekts definiert. Wenn Sie keine Dev-Datei in Ihrem Quell-Repository haben, wird automatisch eine Standard-Dev-Datei angewendet. Die Standard-Devdatei enthält Tools für die am häufigsten verwendeten Programmiersprachen und Frameworks. Standardmäßig ist eine Entwicklungsumgebung so konfiguriert, dass sie über einen 2-Core-Prozessor, 4 GB RAM und 16 GiB persistenten Speicher verfügt.

# Workflows

Ein Workflow ist ein automatisiertes Verfahren, das beschreibt, wie Sie Ihren Code als Teil eines CI/CD-Systems (Continuous Integration and Continuous Delivery) erstellen, testen und bereitstellen. Ein Workflow definiert eine Reihe von Schritten oder Aktionen, die während einer Workflow-Ausführung ausgeführt werden sollen. Ein Workflow definiert auch die Ereignisse oder Auslöser, die den Start des Workflows auslösen. Um einen Workflow einzurichten, erstellen Sie mit dem [visuellen Editor oder dem YAML-Editor](#) der CodeCatalyst Konsole eine Workflow-Definitionsdatei.

## Tip

Um einen schnellen Überblick darüber zu erhalten, wie Sie Workflows in einem Projekt verwenden könnten, [erstellen Sie ein Projekt mit einem Blueprint](#). Jeder Blueprint stellt einen funktionierenden Workflow bereit, den Sie überprüfen, ausführen und mit dem Sie experimentieren können.

Weitere Informationen zu Workflows finden Sie unter [Erstellen, Testen und Bereitstellen mit Workflows in CodeCatalyst](#).

# Aktionen

Eine Aktion ist der Hauptbaustein eines Workflows und definiert eine logische Arbeitseinheit, die während einer Workflow-Ausführung ausgeführt werden soll. In der Regel umfasst ein Workflow mehrere Aktionen, die nacheinander oder parallel ausgeführt werden, je nachdem, wie Sie sie konfiguriert haben.

Weitere Informationen zu Aktionen finden Sie unter [Verwenden von Aktionen](#).

# Problembereiche

Ein Problem ist ein Datensatz, in dem die Arbeit im Zusammenhang mit Ihrem Projekt aufgezeichnet wird. Sie können ein Problem für ein Feature, eine Aufgabe, einen Fehler oder jede andere Arbeit im Zusammenhang mit Ihrem Projekt erstellen. Wenn du agile Entwicklung verwendest, kann ein Problem auch ein Epos oder eine Benutzergeschichte beschreiben.

Weitere Informationen zu Problemen finden Sie unter [Probleme in CodeCatalyst](#).

## Persönliche Zugriffstoken (PATs)

Ein Personal Access Token (PAT) ähnelt einem Passwort. Es ist mit Ihrer Benutzeridentität verknüpft und kann in allen Bereichen und Projekten in verwendet CodeCatalyst werden. Sie verwenden PATs, um auf CodeCatalyst Ressourcen zuzugreifen, die integrierte Entwicklungsumgebungen (IDEs) und Git-basierte Quell-Repositorys umfassen. PATs repräsentieren Sie in CodeCatalyst und Sie können sie in Ihren Benutzereinstellungen verwalten. Ein Benutzer kann mehr als eine PAT haben. Persönliche Zugriffstoken werden nur einmal angezeigt. Es hat sich bewährt, sie sicher auf Ihrem lokalen Computer zu speichern. Standardmäßig laufen PATs nach einem Jahr ab.

Weitere Informationen zu PATs finden Sie unter [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#)

## Rollen

Eine Rolle definiert den Zugriff eines Benutzers auf die Ressourcen für ein Projekt oder einen Bereich und definiert, welche Aktionen dieser Benutzer ausführen kann. Sie wählen die Rolle für einen Benutzer aus, wenn Sie ihn zu einem Projekt einladen. Es gibt Rollen auf Bereichsebene und Rollen auf Projektebene in. CodeCatalyst Ein Benutzer mit einer Administratorrolle auf der richtigen Ebene kann zugewiesene Rollen ändern. Beispielsweise hat ein Benutzer mit der Projektadministratorrolle für ein Projekt die volle Kontrolle über dieses Projekt und kann die Rollen der Benutzer in diesem Projekt ändern. Informationen darüber, welche Rollen verfügbar sind und über welche Berechtigungen die einzelnen Rollen verfügen, finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

Weitere Informationen zu Rollen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).



# Einrichten CodeCatalyst

Es gibt zwei Arten von Bereichen, die Sie einrichten können CodeCatalyst: Bereiche, die AWS Builder ID-Benutzer unterstützen, und die Erstellung eines Bereichs, der den Identitätsverbund unterstützt, in dem SSO-Benutzer und -Gruppen in IAM Identity Center verwaltet werden. Benutzer in einem AWS Builder-ID-Bereich melden sich CodeCatalyst mit ihrer AWS Builder-ID an, und Benutzer in einem Enterprise-Bereich melden sich CodeCatalyst über das SSO-Portal für das Unternehmen an, das dem Bereich zugeordnet ist.

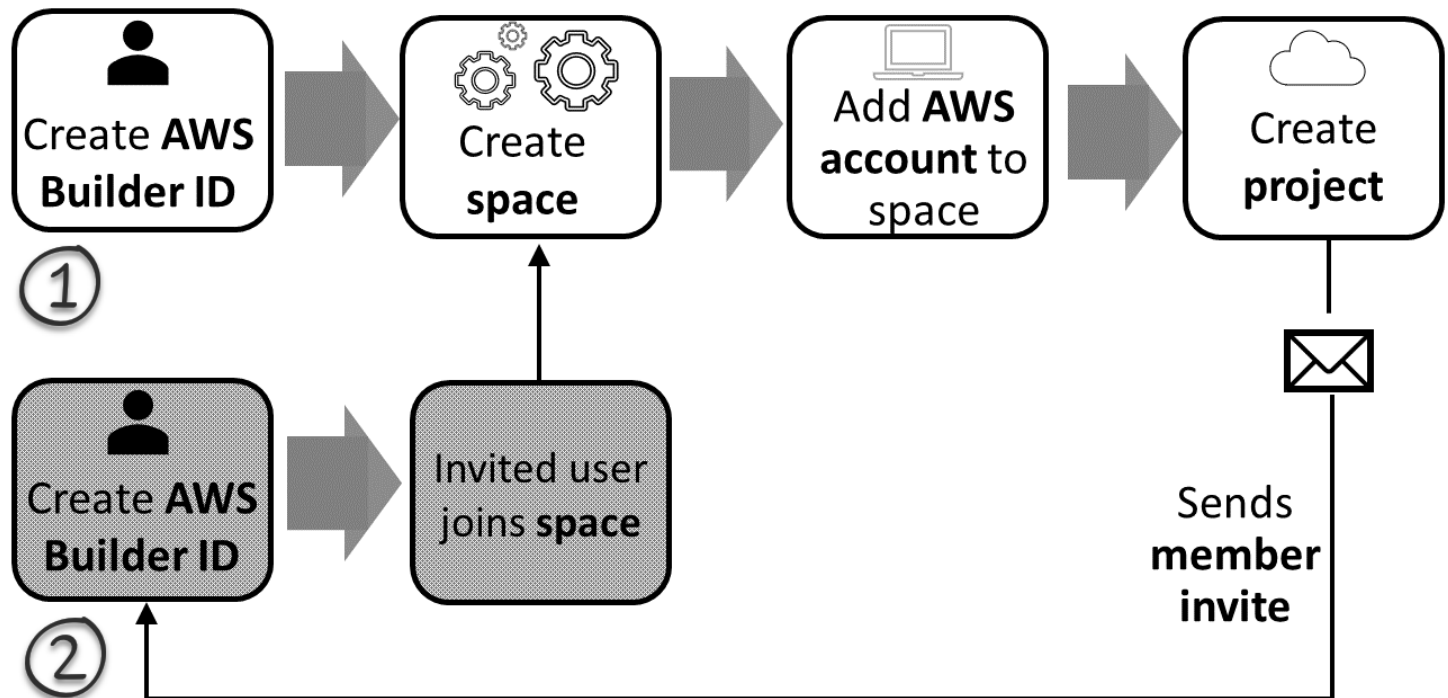
Die Schritte zur Einrichtung und Verwaltung eines AWS Builder-ID-Bereichs werden in diesem Handbuch beschrieben. Um mit einem CodeCatalyst AWS Builder-ID-Bereich zu arbeiten, richten Sie ihn CodeCatalyst anhand der Benutzereinstellungen und der AWS Builder-ID ein, mit der Sie sich CodeCatalyst anmelden.

Die Schritte zum Einrichten und Verwalten eines Bereichs, der den Identitätsverbund unterstützt, finden Sie im CodeCatalyst Administratorhandbuch. Informationen zur Arbeit mit Bereichen, die für den Identitätsverbund eingerichtet sind, finden Sie unter [Einrichtung und Verwaltung von CodeCatalyst Spaces](#) im CodeCatalyst Amazon-Administratorhandbuch.

In diesem Abschnitt finden Sie zwei gängige Methoden für die Einrichtung, um CodeCatalyst mit einem AWS Builder-ID-Bereich in Amazon zu arbeiten: das Erstellen eines Bereichs und eines Projekts als erster Benutzer und das Annehmen einer Einladung zu einem vorhandenen Bereich oder Projekt. Diese Einrichtungs-Workflows sind notwendigerweise sehr unterschiedlich. Das folgende Diagramm zeigt beide Anmeldevorgänge wie folgt:

Es gibt zwei gängige Methoden, um sich für die Arbeit in Amazon einzurichten CodeCatalyst: als erster Benutzer einen Bereich und ein Projekt erstellen und eine Einladung zu einem vorhandenen Bereich oder Projekt annehmen. Diese Einrichtungs-Workflows sind notwendigerweise sehr unterschiedlich. Das folgende Diagramm zeigt beide Anmeldevorgänge wie folgt:

1. Im ersten Fall erstellen und richten Sie einen Bereich für Ihr Unternehmen, Ihr Team oder Ihre Gruppe ein und erstellen ein Projekt, bevor Sie andere zu diesen Ressourcen einladen. Ein AWS-Konto muss für Abrechnungszwecke bereitgestellt werden, wobei Sie weiterhin standardmäßig das kostenlose Kontingent verwenden können.
2. Im zweiten Fall, wenn du beitretest, CodeCatalyst indem du eine Einladung zu einem Projekt annimmst, hat eine andere Person bereits einen Bereich und ein Projekt für dich erstellt. Sie sollten Ihr Profil jedoch trotzdem so konfigurieren, dass Sie bereit sind, mit anderen zusammenzuarbeiten.



**i** Tip

CodeCatalyst verwendet Bereiche, um Projekte und Ressourcen zu gruppieren. Wenn Sie sich zum ersten Mal anmelden CodeCatalyst, werden Sie aufgefordert, sowohl einen Bereich als auch ein Projekt zu erstellen.

Unabhängig davon, ob Sie sich registrieren, um einen Bereich und ein Projekt zu erstellen, oder ob Sie sich im Rahmen der Annahme einer Einladung registrieren, erstellen Sie eine AWS Builder-ID, mit der Sie sich anmelden CodeCatalyst. Um eine AWS Builder-ID zu erstellen, geben Sie den vollständigen Namen, das Passwort und die E-Mail-Adresse an, mit der Sie sich bei AWS Anwendungen anmelden. CodeCatalyst Danach verwenden Sie die E-Mail-Adresse und das Passwort, um sich anzumelden. Sie können diese AWS Builder-ID auch verwenden, um sich bei anderen Anwendungen anzumelden, die AWS Builder-ID-Anmeldeinformationen verwenden.

In CodeCatalyst und in AWS Builder ID wird ein Profil auf der Grundlage Ihrer Anmeldeinformationen generiert. Ihr Profil enthält Ihre CodeCatalyst Präferenzen für Sprach- und Benachrichtigungseinstellungen in Ihren CodeCatalyst Projekten.

**i** Tip

Wenn Sie bei der Registrierung für Ihr CodeCatalyst Amazon-Profil auf Probleme stoßen, folgen Sie den auf dieser Seite angegebenen Schritten. Wenn Sie weitere Hilfe benötigen, finden Sie weitere Informationen unter [Probleme bei der Anmeldung](#).

## Themen

- [Registrieren Sie sich, um Ihren ersten Bereich und Ihre Entwicklungsrolle zu erstellen](#)
- [Eine Einladung annehmen und Ihre AWS Builder-ID erstellen](#)
- [Melden Sie sich mit Ihrer AWS Builder-ID an](#)
- [Nehmen Sie eine E-Mail-Einladung zur Anmeldung mit SSO an](#)
- [Mit SSO anmelden](#)
- [Alle Bereiche und Projekte für einen Benutzer anzeigen](#)
- [CodeCatalyst Profile anzeigen und verwalten](#)
- [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#)

## Registrieren Sie sich, um Ihren ersten Bereich und Ihre Entwicklungsrolle zu erstellen

Sie können sich CodeCatalyst ohne Einladung zu einem vorhandenen Bereich oder Projekt bei Amazon registrieren. Wenn Sie dies tun, erstellen Sie einen Bereich und ein Projekt, nachdem Sie Ihre AWS -Entwickler-ID erstellt haben. Im Rahmen der Erstellung eines Raums müssen Sie einen zu AWS-Konto Fakturierungszwecken hinzufügen.

**i** Tip

Wenn bei der Registrierung für Ihr Amazon- CodeCatalyst Profil Probleme auftreten, führen Sie die auf dieser Seite angegebenen Schritte aus. Weitere Hilfe finden Sie unter [Probleme bei der Anmeldung](#).

Hier ist ein möglicher Ablauf für einen Benutzer, der CodeCatalyst ohne Einladung zu einem Projekt oder einem Bereich mit beginnt.

Mary Major ist ein Entwickler, der sich für ihn interessiert CodeCatalyst und ihn ausprobieren möchte. Sie navigiert zur CodeCatalyst Konsole und wählt die Option zum Registrieren und Erstellen einer AWS -Builder-ID aus. Mary gibt eine E-Mail-Adresse und ein Passwort an, um ihre AWS -Entwickler-ID zu erstellen. Sie wird ihre AWS -Entwickler-ID verwenden können, um sich bei CodeCatalyst und anderen Anwendungen anzumelden. Wenn sie aufgefordert wird, einen Alias auszuwählen, gibt sie MaryMajor als CodeCatalyst Benutzernamen an, der in angezeigt wird CodeCatalyst und den andere Projektmitglieder für @mention Mary verwenden werden.

Als Nächstes wird Mary automatisch angewiesen, einen Raum zu erstellen. Im Rahmen dieses Ablaufs wird Mary aufgefordert, einen AWS-Konto mit dem von ihr erstellten Speicherplatz zu verknüpfen, damit sie den Beispielcode in ihrem ersten Projektaufbau und -bereitstellung sehen kann. Sie fügt diese Informationen hinzu und erstellt ihren Bereich, in dem sie die Option zum Erstellen einer Vorschau-Entwicklungsrolle auswählt, die für Projekte in ihrem neuen Bereich verwendet werden kann. Mary entscheidet sich für die Erstellung eines Projekts und zeigt dann eine Liste von Blueprints für Projekte an. Nachdem sie die Informationen für die verfügbaren Blueprints überprüft hat, entscheidet sie sich, den Blueprint der modernen dreistufigen Webanwendung für ihr erstes Projekt auszuprobieren. Sie füllt die erforderlichen Felder aus und erstellt das Projekt. Sobald das Projekt fertig ist, wird sie zu einer Projektübersichtsseite weitergeleitet, die aktuelle Aktivitäten sowie Links zu Projektcode und dem Workflow enthält, der diesen Code automatisch erstellt und bereitstellt. Sie untersucht sowohl den Code als auch den Workflow, einschließlich der Anzeige der bereitgestellten Beispielwebanwendung. Sie sieht, was sie sieht, und entscheidet sich, einige ihrer Kollegen in das Projekt einzuladen, um mit der Erkundung von zu beginnen CodeCatalyst.


Wenn sie einen Moment hat, konfiguriert Mary ihre AWS Builder-ID so, dass sie sich CodeCatalyst mit Multi-Faktor-Authentifizierung (MFA) bei anmeldet. Wenn MFA konfiguriert ist, kann sich Mary CodeCatalyst mit einer Kombination aus ihrem CodeCatalyst Passwort und einem Passcode oder Token von einer genehmigten Drittanbieter-Authentifizierungs-App bei anmelden.

## Erstellen Ihres ersten Bereichs und Ihrer ersten IAM-Rollen

Gehen Sie wie folgt vor, um sich für Ihr Amazon- CodeCatalyst Profil anzumelden, einen Bereich zu erstellen und ein Konto, eine Supportrolle und eine Entwicklerrolle für Ihren Bereich hinzuzufügen.

Das letzte Verfahren erstellt und fügt die Entwicklerrolle hinzu. Die Entwicklerrolle ist eine AWS IAM-Rolle, die es Ihren CodeCatalyst Workflows ermöglicht, auf -AWSRessourcen zuzugreifen. Die Entwicklerrolle ist eine Servicerolle, die zur Verwaltung von verwendet wird AWS-Services und in dem Konto erstellt wird, das angemeldet ist. Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein

IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Die Rolle hat den Namen `CodeCatalystWorkflowDevelopmentRole-spaceName`. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#).

 Note

Als bewährte Sicherheitsmethode sollten Sie administrativen Zugriff nur Administratorbenutzern und Entwicklern zuweisen, die den Zugriff auf AWS Ressourcen im Bereich verwalten müssen.

Bevor Sie beginnen, müssen Sie bereit sein, eine AWS-Konto ID für ein Konto anzugeben, in dem Sie über Administratorrechte verfügen. Halten Sie Ihre 12-stellige AWS-Konto ID bereit. Informationen zum Suchen Ihrer -AWS-KontoID finden Sie unter [Ihre -AWS-KontoID und der zugehörige Alias](#) .

So melden Sie sich als neuer Benutzer an

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die und stellen Sie sicherAWS Management Console, dass Sie mit derselben angemeldet sindAWS-Konto, die Sie zum Erstellen Ihres Bereichs verwenden möchten.
2. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Wählen Sie auf der Willkommenseite Registrieren aus. Die Seite Erstellen Ihrer AWS - Entwickler-ID wird angezeigt. Ihre AWS -Entwickler-ID ist eine Identität, die Sie für die Anmeldung erstellen. Es ist nicht dasselbe wie ein AWS-Konto.
4. Geben Sie unter Ihre E-Mail-Adresse die E-Mail-Adresse ein, die Sie zuordnen möchten CodeCatalyst. Wählen Sie anschließend Weiter.
5. Geben Sie unter Ihr Name den Vor- und Nachnamen an, den Sie in Anwendungen anzeigen möchten, in denen Sie Ihre AWS -Entwickler-ID verwenden. Leerzeichen sind zulässig. Dies ist Ihr AWS -Entwickler-ID-Profilname, z. B. Mary Major . Sie können den Namen später ändern.  
Wählen Sie Weiter aus. Die Seite E-Mail-Verifizierung wird angezeigt.
6. Ein Verifizierungscode wird an die von Ihnen angegebene E-Mail gesendet. Geben Sie diesen Code in Verifizierungscode ein und wählen Sie dann Überprüfen aus. Wenn Sie Ihren Code nach 5 Minuten nicht erhalten und ihn in Ihren Spam- oder Junk-Ordnern nicht finden können, wählen Sie Code erneut senden aus.

7. Sobald wir Ihren Code verifiziert haben, geben Sie ein Passwort ein, das die Anforderungen unter Passwort und Passwort bestätigen erfüllt.


Aktivieren Sie das Kontrollkästchen, um Ihre Vereinbarung mit der AWS Kundenvereinbarung und den -AWSServicebedingungen zu bestätigen, und wählen Sie dann AWS Builder-ID erstellen aus.

8. Geben Sie auf der Seite CodeCatalyst Alias erstellen einen Alias ein, den Sie für Ihre eindeutige Benutzerkennung in verwenden möchten CodeCatalyst. Wählen Sie eine verkürzte Version Ihres Namens ohne Leerzeichen aus, z. B. MaryMajor. Andere CodeCatalyst Benutzer verwenden dies, um Sie in Kommentaren und Pull-Anfragen zu @mentieren. Ihr CodeCatalyst Profil enthält sowohl Ihren vollständigen Namen aus Ihrer AWS -Entwickler-ID als auch Ihren CodeCatalyst Alias. Sie können Ihren CodeCatalyst Alias später nicht mehr ändern.

Ihr vollständiger Name und Ihr Alias werden in verschiedenen Bereichen in angezeigt CodeCatalyst. Ihr Profilname wird beispielsweise für Ihre aufgeführte Aktivität im Aktivitäts-Feed angezeigt, aber Projektmitglieder verwenden Ihren Alias, um Sie zu @mentieren.

Wählen Sie Weiter aus. Die Seite wird aktualisiert und zeigt den Abschnitt CodeCatalyst Raum erstellen an.

9. Geben Sie unter Benennen Ihres Leerzeichens den Namen Ihres Leerzeichens ein. Sie können dies später nicht mehr ändern.

 Note

Leerzeichennamen müssen in allen eindeutig sein CodeCatalyst. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden.

10. Wählen Sie im AWS-Region Dropdown-Menü die Region aus, in der Sie Ihren Speicherplatz und Ihre Projektdaten speichern möchten. Sie können dies später nicht mehr ändern.
11. Wählen Sie Weiter aus. Die Seite wird aktualisiert und zeigt die Seite zum Hinzufügen eines anAWS-Konto. Dieses Konto wird als Fakturierungskonto für den Speicherplatz verwendet.
12. Geben Sie unter AWS-Konto ID die zwölfstellige ID für das Konto ein, das Sie mit Ihrem Bereich verbinden möchten.

Kopieren Sie im AWS Kontoverifizierungstoken die generierte Token-ID. Das Token wird automatisch für Sie kopiert, aber Sie möchten es möglicherweise speichern, während Sie die AWS Verbindungsanforderung genehmigen.


13. Wählen Sie Zur AWS-Konsole gehen, um zu überprüfen.
14. Die Seite Überprüfen des Amazon- CodeCatalyst Bereichs wird im geöffneten AWS Management Console. Dies ist die Seite Amazon CodeCatalyst Spaces. Möglicherweise müssen Sie sich anmelden, um auf die Seite zugreifen zu können.

Wählen Sie in der dieselbe aus dem AWS Management Console, AWS-Region in der Sie Ihren Speicherplatz erstellen möchten.

Um direkt auf die Seite zuzugreifen, melden Sie sich bei Amazon CodeCatalyst Spaces in der AWS Management Console unter <https://console.aws.amazon.com/codecatalyst/home/> an.

Das Verifizierungstokenfeld in der AWS Management Console wird automatisch mit dem in generierten Token gefüllt CodeCatalyst.

15. (Optional) Wählen Sie unter Autorisierte kostenpflichtige Stufen die Option Bezahlte Stufen autorisieren (Standard, Enterprise), um die kostenpflichtigen Stufen für Ihr Abrechnungskonto zu aktivieren.

 Note

Dadurch wird die Fakturierungsebene nicht auf eine kostenpflichtige Stufe aktualisiert. Dadurch wird jedoch konfiguriert, AWS-Konto sodass Sie die Fakturierungsebene für Ihren Speicherplatz jederzeit in ändern können CodeCatalyst. Sie können die kostenpflichtigen Stufen jederzeit aktivieren. Ohne diese Änderung kann der Speicherplatz nur das kostenlose Kontingent nutzen.

16. Wählen Sie Bereich überprüfen aus.

Es wird eine Meldung zum bestätigten Erfolg des Kontos angezeigt, die anzeigt, dass das Konto dem Bereich hinzugefügt wurde.

17. Bleiben Sie auf der Seite Überprüfen des Amazon- CodeCatalyst Bereichs. Wählen Sie den folgenden Link aus: Um IAM-Rollen für diesen Bereich hinzuzufügen, zeigen Sie Raumdetails an.


Die Seite Verbindungen mit CodeCatalyst Leerzeichen wird im geöffneten AWS Management Console. Dies ist die Seite Amazon CodeCatalyst Spaces. Möglicherweise müssen Sie sich anmelden, um auf die Seite zugreifen zu können.

18. Kehren Sie zur CodeCatalyst Seite zurück und wählen Sie dann Weiter aus.
19. Während der Erstellung Ihres Bereichs wird eine Statusmeldung angezeigt. Wenn der Bereich erstellt wurde, wird CodeCatalyst die folgende Meldung angezeigt: Ihr Bereich ist bereit.



Ihr letzter Schritt ist das Erstellen eines Projekts. Sie können einen der folgenden Schritte ausführen:

- Wählen Sie Für jetzt überspringen aus.
- Wählen Sie Ihr erstes Projekt für Ihren Bereich erstellen aus. Ein Tutorial, das Ihnen zeigt, wie Sie ein Projekt mit einem Blueprint erstellen, finden Sie unter [Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen](#)

 Note

Wenn ein Berechtigungsfehler oder Banner angezeigt wird, aktualisieren Sie die Seite und versuchen Sie erneut, die Seite anzuzeigen.


So erstellen Sie die und fügen sie hinzu CodeCatalyst  
CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die und stellen Sie dann sicherAWS Management Console, dass Sie AWS-Konto für Ihren Speicherplatz mit derselben angemeldet sind.
2. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Navigieren Sie zu Ihrem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
4. Wählen Sie den Link für die ausAWS-Konto, in der Sie die Rolle erstellen möchten. Die AWS-Konto Detailseite wird angezeigt.
5. Wählen Sie Rollen verwalten ausAWS Management Console.

Die Seite IAM-Rolle zum Amazon- CodeCatalyst Bereich hinzufügen wird im geöffnetAWS Management Console. Dies ist die Seite Amazon CodeCatalyst Spaces. Möglicherweise müssen Sie sich anmelden, um auf die Seite zugreifen zu können.

6. Wählen Sie CodeCatalyst Entwicklungsadministratorrolle in IAM erstellen aus. Diese Option erstellt eine Servicerolle, die die Berechtigungsrichtlinie und Vertrauensrichtlinie für die Entwicklungsrolle enthält. Die Rolle hat den Namen CodeCatalystWorkflowDevelopmentRole-*spaceName*. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#).



 Note

Diese Rolle wird nur für die Verwendung mit Entwicklerkonten empfohlen und verwendet die von AdministratorAccess AWS verwaltete Richtlinie, sodass sie vollen Zugriff auf die Erstellung neuer Richtlinien und Ressourcen in dieser hatAWS-Konto.

7. Wählen Sie Entwicklungsrolle erstellen aus.
8. Zeigen Sie auf der Seite Verbindungen unter IAM-Rollen, die für verfügbar sind CodeCatalyst, die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle in der Liste der IAM-Rollen an, die Ihrem Konto hinzugefügt wurden.
9. Um zu Ihrem Bereich zurückzukehren, wählen Sie Zu Amazon gehen aus CodeCatalyst.

So erstellen Sie die und fügen sie hinzu CodeCatalyst AWSRoleForCodeCatalystSupport

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die und stellen Sie dann sicherAWS Management Console, dass Sie AWS-Konto für Ihren Speicherplatz mit derselben angemeldet sind.
2. Navigieren Sie zu Ihrem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
3. Wählen Sie den Link für die ausAWS-Konto, in der Sie die Rolle erstellen möchten. Die AWS-Konto Detailseite wird angezeigt.
4. Wählen Sie Rollen verwalten ausAWS Management Console.

Die Seite IAM-Rolle zum Amazon- CodeCatalyst Bereich hinzufügen wird im geöffnetAWS Management Console. Dies ist die Seite Amazon CodeCatalyst Spaces. Möglicherweise müssen Sie sich anmelden, um auf die Seite zugreifen zu können.

5. Wählen Sie unter CodeCatalyst Raumdetails die Option CodeCatalyst Supportrolle hinzufügen aus. Diese Option erstellt eine Servicerolle, die die Berechtigungsrichtlinie und Vertrauensrichtlinie für die Vorschau-Entwicklungsrolle enthält. Der Rolle wird ein Name AWSRoleForCodeCatalystSupport mit einer eindeutigen Kennung angehängt. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur AWSRoleForCodeCatalystSupport Servicerolle](#).
6. Lassen Sie auf der Seite Rolle für CodeCatalyst Support hinzufügen die Standardeinstellung ausgewählt und wählen Sie dann Rolle erstellen aus.

7. Zeigen Sie unter IAM-Rollen, die für verfügbar sind CodeCatalyst, die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle in der Liste der IAM-Rollen an, die Ihrem Konto hinzugefügt wurden.
8. Um zu Ihrem Bereich zurückzukehren, wählen Sie Zu Amazon gehen aus CodeCatalyst.

Nachdem Sie Ihre AWS -Entwickler-ID erstellt, Ihren ersten Bereich erstellt und ein Konto hinzugefügt haben, können Sie dann ein Projekt erstellen. Weitere Informationen finden Sie unter [Ein Projekt in Amazon erstellen CodeCatalyst](#). Wenn Sie zum ersten Mal verwenden CodeCatalyst, empfehlen wir, mit zu beginnen [Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen](#).

## Eine Einladung annehmen und Ihre AWS Builder-ID erstellen

Sie können sich bei Amazon registrieren CodeCatalyst , wenn Sie eine Einladung zu einem Projekt oder einem Bereich annehmen. Im Rahmen der Annahme der Einladung werden Sie aufgefordert, eine AWS Builder-ID zu erstellen. Sie verwenden Ihre AWS Builder-ID, um auf Ressourcen in zuzugreifen CodeCatalyst.

### Tip

Wenn Sie weitere Hilfe benötigen, finden Sie weitere Informationen unter [Probleme bei der Anmeldung](#).

Hier ist ein möglicher Ablauf für einen Benutzer, der CodeCatalyst mit einer Einladung zu einem Projekt oder einem Bereich beginnt.

Saanvi Sarkar ist ein Entwickler, der eine Einladung erhalten hat, als CodeCatalyst Projektadministrator an einem Projekt teilzunehmen. Saanvi nimmt die Einladung an, wodurch die Anmeldeseite für geöffnet wird. CodeCatalyst Sie entscheidet sich für die Registrierung und gibt eine E-Mail-Adresse und ein Passwort an, um ihre AWS Builder-ID zu erstellen. Saanvi kann sich mit ihrer AWS Builder-ID bei CodeCatalyst und anderen Anwendungen anmelden. Später kann sie ihr Profil bearbeiten, um ihre Login-E-Mail-Adresse oder ihr Passwort zu ändern. Wenn Saanvi aufgefordert wird, einen Alias auszuwählen, gibt er SaanviSarkar als CodeCatalyst Alias an, der in @mention angezeigt wird CodeCatalyst und den andere Projektmitglieder verwenden werden, Saanvi. Nachdem sie sich registriert hat, kann Saanvi ihre Anmeldeinformationen auch für andere Anwendungen verwenden, die Builder-ID-Anmeldeinformationen verwenden. AWS

Nach Abschluss der Registrierung tritt Saanvi automatisch dem CodeCatalyst Projekt und dem Bereich bei, die in der Einladung angegeben sind. Die Einladung enthält auch vorab festgelegte Berechtigungen für ihre Rollen im Projekt und im Space. In den Projekteinstellungen wird Saanvis Alias zusammen mit der ihr zugewiesenen Projektrolle in der Mitgliederliste angezeigt. Um mit Quell-Repositorys zu arbeiten CodeCatalyst, benötigt Saanvi einen Moment, um ein persönliches Zugriffstoken (PAT) zu erstellen. Das PAT wird CodeCatalyst für die Authentifizierung verwendet, wenn Quellenänderungen oder Aktionen vorgenommen werden, für die ein Authentifizierungstoken erforderlich ist.

Wenn Saanvi an einem Projekt arbeitet, wird ihr Alias im Arbeitsaktivitätsprotokoll für das Projekt aufgeführt. Bei Problemen und Kommentaren von Saanvi wird ihr Alias angezeigt, sodass andere Projektmitglieder sie per @mention beantworten können. An @mention, ein anderes Projektmitglied, Saanvi sucht nach seinem Alias in seinem Profil. CodeCatalyst

Wenn sie Zeit hat, konfiguriert Saanvi ihre AWS Builder-ID so, dass sie sich CodeCatalyst mit Multi-Faktor-Authentifizierung (MFA) anmeldet. Wenn MFA konfiguriert ist, kann sich Saanvi CodeCatalyst mit einer Kombination aus ihrem CodeCatalyst Passwort und einem Passcode oder Token aus einer zugelassenen Authentifizierungs-App eines Drittanbieters anmelden.

## Eine Einladung annehmen und eine Builder-ID erstellen AWS

Wenn Sie zu einem Projekt oder Bereich bei Amazon eingeladen werden CodeCatalyst, erhalten Sie eine E-Mail von [notify@codecatalyst.aws](mailto:notify@codecatalyst.aws), in der Sie gebeten werden, die Einladung anzunehmen. Wenn Sie bereits über eine AWS Builder-ID verfügen und angemeldet sind CodeCatalyst, wird das Projekt oder der Bereich automatisch in einem Browser-Tab geöffnet, wenn Sie Einladung annehmen wählen. Wenn Sie nicht bei der Konsole angemeldet sind, aber über eine AWS Builder-ID verfügen, werden Sie zur Anmeldeseite weitergeleitet. Weitere Informationen finden Sie unter [Melden Sie sich mit Ihrer AWS Builder-ID an](#).

Wenn Sie keine AWS Builder-ID haben, gelangen Sie durch Auswahl von Einladung annehmen zur Anmeldeseite, auf der Sie die Option zum Erstellen Ihrer AWS Builder-ID auswählen sollten.

Um eine Einladung anzunehmen und eine AWS Builder-ID zu erstellen

1. Wählen Sie in der Einladungs-E-Mail die Option Einladung annehmen aus.
2. Wählen Sie auf der Anmeldeseite die Option Nicht angemeldet? Erstellen Sie Ihre AWS Builder-ID.

 Tip

Ihre AWS Builder-ID ist eine Identität, die Sie für die Anmeldung erstellen. Es ist nicht dasselbe wie ein AWS-Konto.

3. Geben Sie auf der Seite AWS Builder-ID erstellen im Feld E-Mail-Adresse die E-Mail-Adresse ein, die Sie für Ihre AWS Builder-ID verwenden möchten.

Geben Sie im Feld Ihr Name den Vor- und Nachnamen ein, der in Anwendungen angezeigt werden soll, in denen Sie Ihre AWS Builder-ID verwenden. Leerzeichen sind zulässig. Dies wird Ihr AWS Builder ID-Profilname sein, z. B. Mary Major. Sie können den Namen später ändern.

Wählen Sie Weiter.

Ein Bestätigungscode wird an die von Ihnen angegebene E-Mail-Adresse gesendet. Geben Sie diesen Code in das Feld Bestätigungscode ein und wählen Sie dann Verifizieren aus. Wenn du deinen Code nach 5 Minuten nicht erhältst und ihn auch nicht in deinen Spam- oder Junk-Ordern finden kannst, wähle Code erneut senden.

4. Sobald dein Code verifiziert wurde, gib ein Passwort ein, das den Anforderungen unter Passwort und Passwort bestätigen entspricht.
5. Wählen Sie Create AWS Builder ID.
6. Geben Sie auf der Seite „Ihren Alias erstellen“ einen Alias ein, den Sie für Ihre eindeutige Benutzer-ID verwenden möchten CodeCatalyst. Wählen Sie eine verkürzte Version Ihres Namens ohne Leerzeichen, wie MaryMajorz. Andere CodeCatalyst Benutzer werden dies verwenden, um Sie in Kommentaren und Pull-Requests zu @mention zu kontaktieren. Ihr CodeCatalyst Profil wird sowohl Ihren vollständigen Namen aus Ihrer AWS Builder-ID als auch Ihren CodeCatalyst Alias enthalten. Sie können Ihren CodeCatalyst Alias nicht ändern.

Ihr vollständiger Name und Ihr Alias werden in verschiedenen Bereichen von angezeigt CodeCatalyst. Ihr Profilname wird beispielsweise für Ihre aufgelistete Aktivität im Aktivitätsfeed angezeigt, aber die Projektmitglieder verwenden Ihren Alias, um Sie zu @mention zu kontaktieren.

Wählen Sie Alias erstellen aus. Du wirst zu dem Projekt oder Space weitergeleitet, zu dem du eingeladen wurdest.

## Melden Sie sich mit Ihrer AWS Builder-ID an

Folgen Sie diesen Schritten, um sich bei Ihrem CodeCatalyst Amazon-Profil anzumelden.

### Note

Haben Sie schon ein Gerät für die Multi-Faktor-Authentifizierung (MFA) registriert? Wir empfehlen dringend, MFA in Amazon CodeCatalyst zu konfigurieren, um Ihre Sicherheit zu erhöhen. Weitere Informationen finden Sie unter [Wie registriere ich ein Gerät für die Verwendung mit Multi-Faktor-Authentifizierung](#).

So melden Sie sich mit Ihrer AWS-Entwickler-ID an

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Geben Sie Ihre E-Mail-Adresse ein. Wählen Sie optional Meine E-Mail-Adresse speichern, wenn Sie Ihre E-Mail-Adresse für future Anmeldungen speichern möchten. Klicken Sie auf Weiter.
3. Geben Sie Ihr Passwort (Passwort) ein. Klicken Sie auf Sign in. Wenn Sie sich nicht an Ihr Passwort erinnern können, folgen Sie den Schritten unter [Ich habe mein Passwort vergessen](#)

## Vertrauenswürdige Geräte

Nachdem Sie auf der Anmeldeseite die Option Dies ist ein vertrauenswürdige Gerät ausgewählt haben, CodeCatalyst betrachtet Amazon alle future Anmeldungen von diesem Gerät aus als autorisiert. Amazon CodeCatalyst bietet keine Option zur Eingabe eines MFA-Code an, solange Sie dieses vertrauenswürdige Gerät verwenden. Zu einigen Ausnahmen gehören die Anmeldung über einen neuen Browser oder wenn Ihrem Gerät eine unbekannte IP-Adresse zugewiesen wurde.

## Nehmen Sie eine E-Mail-Einladung zur Anmeldung mit SSO an

Benutzer, die zu einem Bereich hinzugefügt werden, der den Identitätsverbund unterstützt, erhalten eine E-Mail mit dem Link zum Anmeldeportal und den Einrichtungsinformationen. Gehen Sie wie folgt vor, um eine Einladung anzunehmen und sich mit SSO anzumelden.

Informationen zur Anmeldung stattdessen mit Ihrer AWS Builder-ID finden Sie unter [Melden Sie sich mit Ihrer AWS Builder-ID an](#).

So akzeptieren Sie SSO und melden sich mit SSO an

1. Wählen Sie die Schaltfläche in der E-Mail, mit der die Anfrage akzeptiert wird. Verwenden Sie den Link in der E-Mail, um zum Anmeldeportal für den Bereich zu gelangen, der Ihrem Unternehmen zugeordnet ist.
2. Geben Sie unter Nutzernamen und Passwort Ihre Anmeldedaten ein.
3. Klicken Sie auf Sign in.

## Mit SSO anmelden

Gehen Sie wie folgt vor, um sich mit SSO bei Amazon anzumelden CodeCatalyst.

Informationen zur Anmeldung stattdessen mit Ihrer AWS Builder-ID finden Sie unter [Melden Sie sich mit Ihrer AWS Builder-ID an](#).

So melden Sie sich mit SSO an

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie unter Wählen Sie eine Anmeldeoption die Option Single Sign-On (SSO) verwenden aus.
3. Geben Sie im Feld AWS Identity Center-Anwendungsname den Anwendungsnamen ein, den Sie von Ihrem Identity Federation-Administrator erhalten haben.
4. Wählen Sie Weiter zu IAM Identity Center.

## Alle Bereiche und Projekte für einen Benutzer anzeigen

Sie können eine Liste Ihrer Bereiche und Projekte auf der Benutzerhomepage einsehen. Die Benutzerhomepage enthält eine Liste aller Bereiche, denen der Benutzer angehört, die Rolle des Benutzers in diesem Bereich, z. B. Space-Administrator, und die Projekte in jedem Bereich, in dem der Benutzer Mitglied ist.

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Geben Sie im Browser die folgende Adresse ein: <https://codecatalyst.aws/home>

The screenshot displays the Amazon CodeCatalyst interface. At the top, there is a header bar with the text 's spaces (9)' on the left, a 'Manage AWS Builder ID' button with an external link icon in the center, and a 'Create space' button on the right. Below the header is a search bar labeled 'Filter spaces'. The main content area is divided into three sections, each representing a different space:

- EnchantedForest**: Space administrator. A 'Create project' button is visible. Below this, there are two project cards: 'WildWaves' and 'FracturedFairyTales'. Each card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'.
- org**: Space administrator. A 'Create project' button is visible. Below this, there are four project cards: 'migration', 'test', and '12597'. Each card lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'.
- AnyCompany**: Space member. A 'Create project' button is visible. Below this, there is one project card: 'newproject', which lists 'Pull requests', 'Workflows', 'Source repositories', and 'Environments'.

3. Wählen Sie den Bereich oder das Projekt aus, den Sie öffnen möchten. Wenn Sie einen Bereich oder ein Projekt nicht sehen, den Sie erwartet haben, müssen Sie sich möglicherweise als ein anderer Benutzer anmelden.

## CodeCatalyst Profile anzeigen und verwalten

Sie können Benutzerprofile in Amazon einsehen CodeCatalyst , um Informationen wie E-Mail-Adressen und CodeCatalyst Aliase zu erhalten. Sie können auch Ihr Profil und Ihre AWS Builder-ID

aktualisieren. Wenn Sie Ihr Passwort vergessen haben, können Sie ein Zurücksetzen des Passworts beantragen.

## Ihr CodeCatalyst Profil ansehen

Sie geben bei der Registrierung Informationen an, die als Ihre Anmeldeinformationen für die Anmeldung bei Amazon verwendet CodeCatalyst und in Ihrem Profil verwaltet werden. Dazu gehören Ihr Name, Ihr Spitzname und die E-Mail-Adresse, mit der Sie sich CodeCatalyst anmelden.

### Note

Der AWS Builder-ID-Spitzname ist nicht Ihr CodeCatalyst Alias. Sie haben Ihren CodeCatalyst Alias bei der Anmeldung ausgewählt.

Um dein CodeCatalyst Profil anzusehen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Klicken Sie rechts oben auf den Pfeil neben dem Symbol mit Ihrer ersten Initiale und wählen Sie dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.
3. Um Ihre AWS Builder-ID-E-Mail-Adresse oder Ihr Passwort zu aktualisieren oder MFA einzurichten, wählen Sie AWSBuilder-ID verwalten. Die AWS Builder-ID-Seite wird geöffnet.

## Das CodeCatalyst Profil eines anderen Benutzers anzeigen

Um das CodeCatalyst Profil eines anderen Benutzers anzusehen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie in der Seitennavigation die Option Projekteinstellungen aus. Wählen Sie den Tab Mitglieder. Sehen Sie sich die Liste der Mitglieder für Ihr CodeCatalyst Projekt an.
3. Wählen Sie den Mitgliedsnamen, nach dem Sie suchen möchten, oder @mention. Auf der Seite Meine Einstellungen werden der Alias, die E-Mail-Adresse und der vollständige Name des Benutzers angezeigt. Verwenden Sie den CodeCatalyst Alias to @mention -Projektmitglieder.



**Note**

Der AWS Builder-ID-Spitzname eines Benutzers ist nicht sein CodeCatalyst Alias. Sie haben ihren CodeCatalyst Alias bei der Anmeldung ausgewählt.

Um das Profil eines anderen Benutzers in Ihrem Projekt anzuzeigen, wählen Sie dessen Namen in der Liste aus.

## Aktualisierung Ihres Profils

Ihr Profil besteht aus persönlichen Informationen CodeCatalyst, die von AWSBuilder ID verwaltet werden, und Einstellungen, die unter verwaltet werden CodeCatalyst.

- Der vollständige Name, die E-Mail-Adresse und das Passwort Ihres Profils werden von AWSBuilder ID verwaltet. Sie haben diese Informationen bei der Registrierung eingegeben. Wenn Sie MFA für die Verwendung einer Authenticator-App für die Anwendungsanmeldung einrichten, CodeCatalyst gelangen Sie zur AWSBuilder-ID-Seite.
- CodeCatalyst Die Einstellungen für Ihr persönliches Zugriffstoken (PAT), CodeCatalyst Benachrichtigungen und Spracheinstellungen werden auf der Seite Meine Einstellungen unter verwaltet. CodeCatalyst Weitere Informationen finden Sie unter [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#).

**Note**


Sie können den vollständigen Namen (CodeCatalyst Anzeigename) und den Vornamen Ihrer AWS Builder-ID aktualisieren. Sie können Ihren CodeCatalyst Alias jedoch nicht ändern.

## Aktualisierung Ihrer AWS Builder-ID oder E-Mail-Adresse

Um Ihre AWS Builder ID oder Ihre E-Mail-Adresse zu aktualisieren


1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Klicken Sie rechts oben auf den Pfeil neben dem Symbol mit Ihrer ersten Initiale und wählen Sie dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.

3. Wählen Sie auf der Profilseite die Option AWSBuilder-ID verwalten aus. Die AWSBuilder-ID-Seite wird geöffnet.
4. Wählen Sie links auf der Seite Meine Daten aus.
5. Wählen Sie unter Profilinformationen die Option Bearbeiten aus, um Ihren Namen oder Spitznamen zu aktualisieren. Wenn Sie keinen Spitznamen angegeben haben, spiegelt das Feld Spitzname den Vornamen im vollständigen Namen wider. Es ist nicht dein CodeCatalyst Alias.

 Note

Dadurch werden der vollständige Name und der Vorname der AWS Builder-ID aktualisiert. Ihr CodeCatalyst Alias wird dadurch nicht aktualisiert.

Wählen Sie unter Kontaktinformationen die Option Bearbeiten aus, um Ihre E-Mail-Adresse zu aktualisieren.

 Note

Dadurch wird die E-Mail-Adresse aktualisiert, mit der Sie sich anmelden werden CodeCatalyst.

## Dein CodeCatalyst Passwort ändern

Um dein CodeCatalyst Passwort zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Klicken Sie rechts oben auf den Pfeil neben dem Symbol mit Ihrer ersten Initialie und wählen Sie dann Benutzerprofil aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.
3. Wählen Sie auf der Profilseite die Option AWSBuilder-ID verwalten aus. Die AWS Builder-ID-Seite wird geöffnet.
4. Wählen Sie links auf der Seite Sicherheit aus.
5. Wählen Sie Passwort ändern und folgen Sie den Anweisungen.

# Einrichtung für die AWS CLI Verwendung von CodeCatalyst

In der CodeCatalyst Amazon-Konsole werden Sie an den meisten Ihrer täglichen Aufgaben arbeiten. Möglicherweise möchten Sie jedoch das einrichten und konfigurieren, AWS CLI wenn Sie mit Entwicklungsumgebungen, persönlichen Zugriffstoken oder Protokollen von Ereignissen arbeiten CodeCatalyst. Sie müssen das Profil installieren AWS CLI und konfigurieren, bevor Sie es mit verwenden können CodeCatalyst.

Um das einzurichten AWS CLI für CodeCatalyst

1. Installieren Sie die neueste Version von AWS CLI. Wenn Sie bereits eine Version von AWS CLI installiert haben, stellen Sie sicher, dass sie aktuell ist und Befehle für enthält CodeCatalyst, und aktualisieren Sie sie bei Bedarf. Um zu überprüfen, ob Sie eine Version installiert haben, die CodeCatalyst Befehle enthält, öffnen Sie eine Befehlszeile und führen Sie den folgenden Befehl aus:

```
aws codecatalyst help
```

Wenn Sie eine Liste mit CodeCatalyst Befehlen sehen, haben Sie eine Version, die diese Befehle unterstützt CodeCatalyst. Wenn der Befehl nicht erkannt wird, aktualisieren Sie Ihre Version von AWS CLI auf die neueste Version. Weitere Informationen finden Sie im AWS Command Line Interface Benutzerhandbuch unter [Installation oder Aktualisierung AWS CLI der neuesten Version von](#).

2. Führen Sie den `aws configure` Befehl aus, um ein Profil zu erstellen, falls Sie noch keines haben oder wenn Sie ein benanntes Profil speziell für verwenden möchten CodeCatalyst. Wir empfehlen, ein benanntes Profil für die spezielle Verwendung zu erstellen CodeCatalyst, aber Sie können auch das Standardprofil verwenden. Weitere Informationen finden Sie unter [Grundlagen der Konfiguration](#).
3. Bearbeiten Sie die `config` Datei für das Profil, um einen Abschnitt hinzuzufügen, mit dem Sie eine Verbindung herstellen können, CodeCatalyst wie folgt. Die Datei `config` befindet sich in `~/.aws/config` unter Linux und in macOS oder in `C:\Users\USERNAME\.aws\config` unter 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. Speichern Sie die Datei.
5. Bevor Sie versuchen, CodeCatalyst Befehle auszuführen, öffnen Sie ein neues Terminal oder eine neue Befehlszeile und führen Sie den folgenden Befehl aus, um Anmeldeinformationen für die Ausführung von `aws codecatalyst` Befehlen anzufordern und abzurufen. Ersetzen Sie es `codecatalyst` bei Bedarf durch den Namen Ihres Profils.

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

Beispiele für `codecatalyst` Befehle finden Sie in den folgenden Themen:

- [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#)
- [Zugreifen auf protokollierte Ereignisse in CodeCatalyst](#)

# Tutorials für die ersten Schritte

Amazon CodeCatalyst bietet eine Reihe verschiedener Vorlagen, um Ihnen den Einstieg in Ihre Projekte zu erleichtern. Sie können sich auch dafür entscheiden, mit einem leeren Projekt zu beginnen und diesem Ressourcen hinzuzufügen. Folgen Sie den Schritten in diesen Tutorials, um zu erfahren, wie Sie damit arbeiten können CodeCatalyst.

Wenn Sie es zum ersten Mal verwenden CodeCatalyst, empfehlen wir, mit zu beginnen [Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen](#).

## Note

Um diesen Tutorials folgen zu können, müssen Sie zunächst die Einrichtung abschließen. Weitere Informationen finden Sie unter [Einrichten CodeCatalyst](#).

## Themen

- [Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen](#)
- [Tutorial: Mit einem leeren Projekt beginnen und Ressourcen manuell hinzufügen](#)
- [Tutorial: Verwenden CodeCatalyst generativer KI-Funktionen zur Beschleunigung Ihrer Entwicklungsarbeit](#)
- [Tutorial: Erstellen einer Full-Stack-Anwendung mit zusammensetzbaren PDK-Blueprints](#)

Weitere Tutorials, die sich auf bestimmte Funktionsbereiche konzentrieren CodeCatalyst, finden Sie unter:

- [Erste Schritte mit Slack-Benachrichtigungen](#)
- [Erste Schritte mit CodeCatalyst Quell-Repositorys und dem Single-Page-Anwendungs-Blueprint](#)
- [Erste Schritte mit Workflows in CodeCatalyst](#)
- [Erste Schritte mit benutzerdefinierten Blueprints](#)
- [Erste Schritte mit dem Amazon CodeCatalyst Action Developer Guide](#)

Ausführliche Tutorials finden Sie unter:

- [Tutorial: Artefakte auf Amazon S3 hochladen](#)

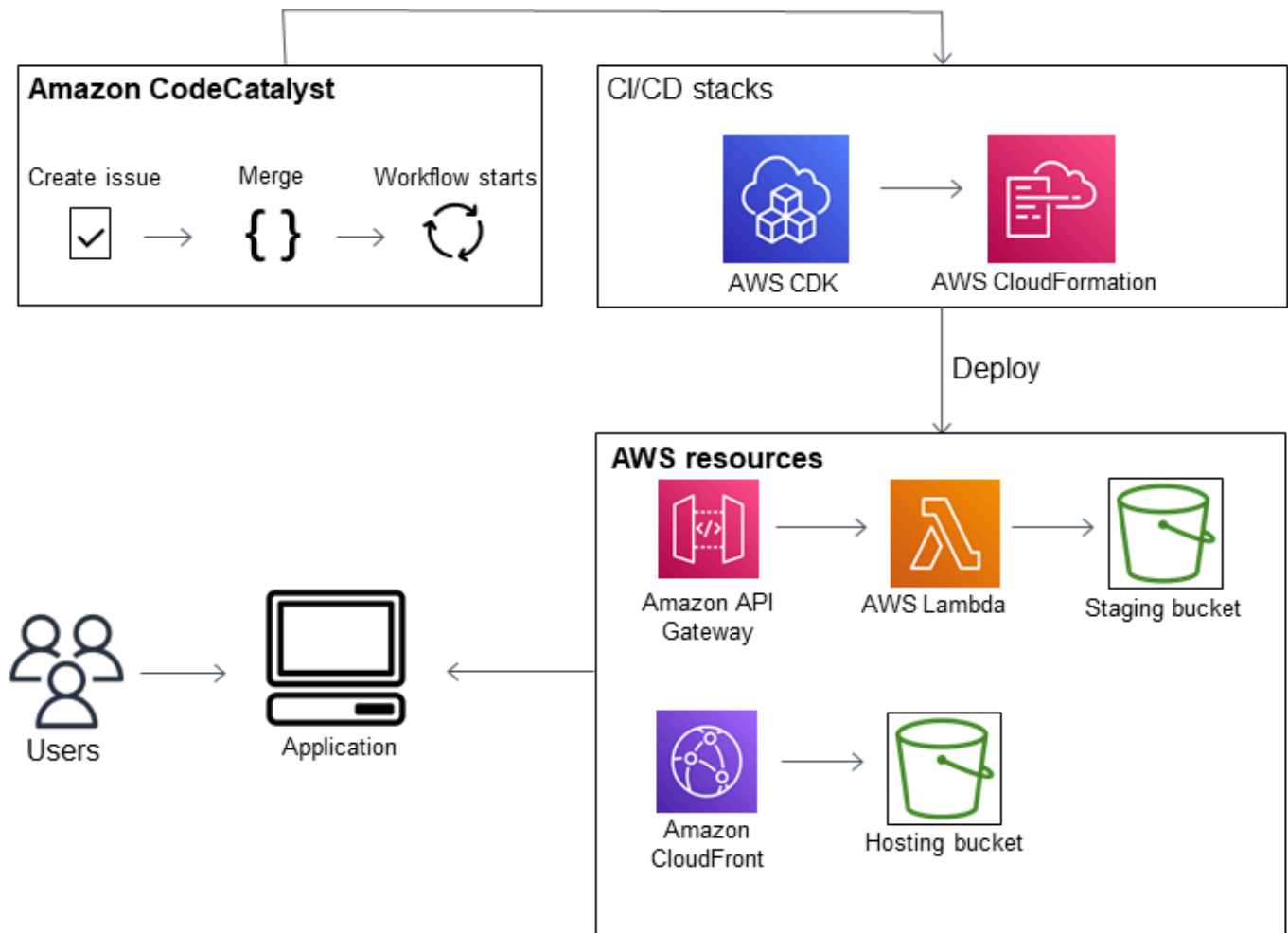
- [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#)
- [Tutorial: Bereitstellen einer Anwendung in Amazon ECS](#)
- [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#)
- [Tutorial: Lint-Code mit einer GitHub Aktion](#)
- [Tutorial: Eine React-Anwendung erstellen und aktualisieren](#)

## Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen

Sie können schneller mit der Softwareentwicklung beginnen, indem Sie ein Projekt mit einem Blueprint erstellen. Ein mit einem Blueprint erstelltes Projekt umfasst die Ressourcen, die Sie benötigen, einschließlich eines Quell-Repositorys zur Verwaltung Ihres Codes und eines Workflows zum Erstellen und Bereitstellen der Anwendung. In diesem Tutorial zeigen wir Ihnen, wie Sie mithilfe des Blueprints für moderne dreistufige Webanwendungen ein Projekt in Amazon erstellen. CodeCatalyst Das Tutorial beinhaltet auch das Anzeigen des bereitgestellten Beispiels, das Einladen anderer Benutzer, daran zu arbeiten, und das Vornehmen von Änderungen am Code mit Pull-Requests, die automatisch erstellt und für Ressourcen im verbundenen Netzwerk bereitgestellt werden, AWS-Konto wenn der Pull-Request zusammengeführt wird. Wo Ihr Projekt mit Berichten, Aktivitätsfeeds und anderen Tools CodeCatalyst erstellt wird, erstellt Ihr Blueprint AWS Ressourcen in den mit Ihrem Projekt AWS-Konto verknüpften Bereichen. Ihre Blueprint-Dateien ermöglichen es Ihnen, eine moderne Beispielanwendung zu erstellen und zu testen und sie in der Infrastruktur in der bereitzustellen. AWS Cloud

Die folgende Abbildung zeigt, wie Tools in verwendet CodeCatalyst werden, um ein Problem für die Nachverfolgung zu erstellen, die Änderung zusammenzuführen und automatisch zu erstellen und dann einen Workflow im CodeCatalyst Projekt zu starten, der Aktionen ausführt, um Ihre Infrastruktur AWS CloudFormation zu ermöglichen AWS CDK und bereitzustellen.

Die Aktionen generieren Ressourcen in den zugehörigen Dateien AWS-Konto und stellen Ihre Anwendung für eine serverlose AWS Lambda Funktion mit einem API-Gateway-Endpunkt bereit. Die AWS Cloud Development Kit (AWS CDK) Aktion konvertiert einen oder mehrere AWS CDK Stapel in AWS CloudFormation Vorlagen und stellt Stapel in Ihren bereit. AWS-Konto Zu den Ressourcen in Ihren Stacks gehören CloudFront Amazon-Ressourcen zur Verteilung dynamischer Webinhalte, eine Amazon DynamoDB-Instance für Ihre Anwendungsdaten sowie die Rollen und Richtlinien, die die bereitgestellte Anwendung unterstützen.



Wenn Sie ein Projekt mit dem Blueprint für moderne dreistufige Webanwendungen erstellen, wird Ihr Projekt mit den folgenden Ressourcen erstellt:

Im Projekt: CodeCatalyst

- Ein [Quell-Repository](#) mit Beispielcode und Workflow-YAML
- Ein [Workflow](#), der den Beispielcode erstellt und bereitstellt, wenn eine Änderung am Standardzweig vorgenommen wird
- Ein Issue-Board und ein Backlog, das Sie zur Planung und Nachverfolgung der Arbeit verwenden können
- Eine Testberichts-Suite mit automatisierten Berichten, die im Beispielcode enthalten sind

Im zugehörigen AWS-Konto:

- Drei AWS CloudFormation Stapel, die die für die Anwendung benötigten Ressourcen erstellen.

Ausführlichere Informationen zu den Ressourcen, die in AWS und CodeCatalyst als Teil dieses Tutorials erstellt werden, finden Sie unter [Referenz](#).

#### Note

Welche Ressourcen und Beispiele in einem Projekt enthalten sind, hängt davon ab, welchen Blueprint Sie auswählen. Amazon CodeCatalyst bietet mehrere Projekt-Blueprints an, die Ressourcen in Bezug auf ihre definierte Sprache oder ihr Framework definieren. Weitere Informationen zu Blueprints finden Sie unter [Referenz zum Projekt-Blueprint](#)

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie das moderne dreistufige Webanwendungsprojekt](#)
- [Schritt 2: Laden Sie jemanden zu Ihrem Projekt ein](#)
- [Schritt 3: Erstelle Aufgaben, an denen du gemeinsam arbeiten kannst, und verfolge die Arbeit](#)
- [Schritt 4: Sehen Sie sich Ihr Quell-Repository an](#)
- [Schritt 5: Erstellen Sie eine Entwicklungsumgebung mit einem Testzweig und nehmen Sie eine schnelle Codeänderung vor](#)
- [Schritt 6: Sehen Sie sich den Workflow an, der die moderne Anwendung erstellt](#)
- [Schritt 7: Bitten Sie andere, Ihre Änderungen zu überprüfen](#)
- [Schritt 8: Schließen Sie das Problem](#)
- [Bereinigen von -Ressourcen](#)
- [Referenz](#)

## Voraussetzungen

Um in diesem Tutorial ein modernes Anwendungsprojekt zu erstellen, müssen Sie die folgenden Aufgaben erledigt haben: [Einrichten CodeCatalyst](#)

- Sie müssen über eine AWS Builder-ID für die Anmeldung verfügen CodeCatalyst.



- Gehören Sie einem Bereich an und lassen Sie sich in diesem Bereich die Rolle Space-Administrator oder Poweruser zuweisen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#), [Verwaltung von Space-Benutzern](#) und [Rolle „Space-Administrator“](#).
- Ordnen Sie Ihrem AWS-Konto Bereich eine zu und lassen Sie sich die IAM-Rolle zuweisen, die Sie bei der Registrierung erstellt haben. Bei der Registrierung haben Sie beispielsweise die Möglichkeit, eine Servicerolle mit einer Rollenrichtlinie zu erstellen, die CodeCatalystWorkflowDevelopmentRole-*spaceName* als Rollenrichtlinie bezeichnet wird. Der Rolle wird ein Name CodeCatalystWorkflowDevelopmentRole-*spaceName* mit einer angehängten eindeutigen Kennung zugewiesen. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#). Die Schritte zum Erstellen der Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#).

## Schritt 1: Erstellen Sie das moderne dreistufige Webanwendungsprojekt

Nachdem Sie es erstellt haben, werden Sie in Ihrem Projekt Code entwickeln und testen, Entwicklungsaufgaben koordinieren und Projektmetriken einsehen. Ihr Projekt enthält auch Ihre Entwicklungstools und Ressourcen.


In diesem Tutorial verwenden Sie den Blueprint für moderne dreistufige Webanwendungen, um eine interaktive Anwendung zu erstellen. Der Workflow, der im Rahmen Ihres Projekts automatisch erstellt und ausgeführt wird, dient der Erstellung und Bereitstellung der Anwendung. Der Workflow wird erst erfolgreich ausgeführt, nachdem alle Rollen und Kontoinformationen für Ihren Bereich konfiguriert wurden. Nachdem der Workflow erfolgreich ausgeführt wurde, können Sie die Endpunkt-URL aufrufen, um die Anwendung zu sehen.

Um ein Projekt mit einem Blueprint zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
3. Wählen Sie Create project (Projekt erstellen) aus.
4. Wählen Sie Mit einem Blueprint beginnen aus.
5. Geben Sie im Suchfeld **modern** ein.
6. Wählen Sie den Blueprint „Moderne dreistufige Webanwendung“ und dann „Weiter“ aus.

7. Geben Sie im Feld Name Ihres Projekts einen Projektnamen ein. Beispielsweise:

**MyExampleProject.**

 Note

Der Name muss in Ihrem Bereich eindeutig sein.

8. Wählen Sie unter Konto den aus, den AWS-Konto Sie bei der Registrierung hinzugefügt haben. Der Blueprint installiert Ressourcen in diesem Konto.
9. Wählen Sie unter Deployment Role die Rolle aus, die Sie bei der Registrierung hinzugefügt haben. Wählen Sie zum Beispiel aus `CodeCatalystWorkflowDevelopmentRole-spaceName`.

Wenn keine Rollen aufgeführt sind, fügen Sie eine hinzu. Um eine Rolle hinzuzufügen, wählen Sie IAM-Rolle hinzufügen und fügen Sie die Rolle zu Ihrer AWS-Konto hinzu. Weitere Informationen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

10. Wählen Sie unter Compute Platform die Option Lambda aus.
11. Wählen Sie in der Frontend-Hosting-Option Amplify Hosting aus. Weitere Informationen dazu finden Sie AWS Amplify unter [Was ist AWS Amplify Hosting?](#) im AWS Amplify Benutzerhandbuch.
12. Geben Sie unter Bereitstellungsregion den Regionalcode der Region ein, AWS-Region in der der Blueprint die Mysfits-Anwendung und die unterstützenden Ressourcen bereitstellen soll. Eine Liste der Regionscodes finden Sie unter [Regionale Endpunkte](#) in der. Allgemeine AWS-Referenz
13. Behalten Sie im Feld Anwendungsname den Standardwert von bei. `mysfitsstring`
14. (Optional) Wählen Sie unter Projektvorschau generieren die Option Code anzeigen aus, um eine Vorschau der Quelldateien anzuzeigen, die mit dem Blueprint installiert werden. Wählen Sie Workflow anzeigen aus, um eine Vorschau der CI/CD-Workflow-Definitionsdateien anzuzeigen, die der Blueprint installieren wird. Die Vorschau wird basierend auf Ihrer Auswahl dynamisch aktualisiert.
15. Wählen Sie Create project (Projekt erstellen) aus.

Der Projekt-Workflow beginnt, sobald Sie das Projekt erstellen. Es wird einige Zeit dauern, bis der Code fertig erstellt und bereitgestellt ist. Laden Sie in der Zwischenzeit eine andere Person zu Ihrem Projekt ein.

## Schritt 2: Laden Sie jemanden zu Ihrem Projekt ein

Nachdem Sie Ihr Projekt eingerichtet haben, laden Sie andere ein, mit Ihnen zu arbeiten.

Um jemanden zu Ihrem Projekt einzuladen

1. Navigieren Sie zu dem Projekt, zu dem Sie Benutzer einladen möchten.
2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Wählen Sie auf der Registerkarte Mitglieder die Option Einladen aus.
4. Geben Sie die E-Mail-Adressen der Personen ein, die Sie als Benutzer für Ihr Projekt einladen möchten. Sie können mehrere E-Mail-Adressen eingeben, die durch ein Leerzeichen oder ein Komma getrennt sind. Sie können auch Mitglieder Ihres Bereichs auswählen, die nicht Mitglieder des Projekts sind.
5. Wählen Sie die Rolle für den Benutzer aus.

Wenn Sie mit dem Hinzufügen von Benutzern fertig sind, wählen Sie Einladen.

## Schritt 3: Erstelle Aufgaben, an denen du gemeinsam arbeiten kannst, und verfolge die Arbeit

CodeCatalyst hilft Ihnen dabei, Funktionen, Aufgaben, Fehler und alle anderen Arbeiten im Zusammenhang mit Problemen im Zusammenhang mit Ihrem Projekt nachzuverfolgen. Sie können Probleme erstellen, um die benötigte Arbeit und Ideen nachzuverfolgen. Wenn du ein Problem erstellst, wird es standardmäßig zu deinem Backlog hinzugefügt. Du kannst Probleme in ein Board verschieben, in dem du die laufenden Arbeiten nachverfolgen kannst. Du kannst ein Problem auch einem bestimmten Projektmitglied zuweisen.

Um ein Problem für ein Projekt zu erstellen

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie Problem erstellen aus.
3. Geben Sie im Feld Titel der Ausgabe einen Namen für das Problem ein. Geben Sie optional eine Beschreibung des Problems an. Verwenden Sie in diesem Beispiel **make a change in the src/myfit\_data.json file**.
4. Wählen Sie die Priorität, die Schätzung, den Status und die Bezeichnungen aus. Wählen Sie unter Beauftragter die Option +Add me aus, um das Problem Ihnen selbst zuzuweisen.

5. Wählen Sie „Problem erstellen“. Das Problem ist jetzt auf der Pinnwand sichtbar. Wähle die Karte aus, um das Problem in die Spalte In Bearbeitung zu verschieben.

Weitere Informationen finden Sie unter [Probleme in CodeCatalyst](#).

## Schritt 4: Sehen Sie sich Ihr Quell-Repository an

Ihr Blueprint installiert ein Quell-Repository, das Dateien zur Definition und Unterstützung Ihrer Anwendung oder Ihres Dienstes enthält. Einige bemerkenswerte Verzeichnisse und Dateien im Quell-Repository sind:

- `.cloud9`-Verzeichnis — Enthält unterstützende Dateien für die AWS Cloud9 Entwicklungsumgebung.
- `.codecatalyst`-Verzeichnis — Enthält die YAML Workflow-Definitionsdatei für jeden im Blueprint enthaltenen Workflow.
- `.idea`-Verzeichnis — Enthält unterstützende Dateien für die Entwicklungsumgebungen. JetBrains
- `.vscode`-Verzeichnis — Enthält unterstützende Dateien für die Visual Studio Code-Entwicklungsumgebung.
- Verzeichnis `cdkStacks` — Enthält die AWS CDK Stack-Dateien, die die Infrastruktur in der definieren. AWS Cloud
- Verzeichnis `src` — Enthält den Quellcode der Anwendung.
- Testverzeichnis — Enthält Dateien für die Integ- und Unit-Tests, die als Teil des automatisierten CI/CD-Workflows ausgeführt werden, der beim Erstellen und Testen Ihrer Anwendung ausgeführt wird.
- Webverzeichnis — Enthält den Frontend-Quellcode. Andere Dateien enthalten Projektdateien wie die `package.json` Datei, die wichtige Metadaten zu Ihrem Projekt enthält, die `index.html` Seite für die Website, die `.eslintrc.cjs` Datei für den Lint-Code und die `tsconfig.json` Datei zur Angabe von Stammdateien und Compileroptionen.
- `Dockerfile` Datei — Beschreibt den Container der Anwendung.
- `README.md` Datei — Enthält Konfigurationsinformationen für das Projekt.

Um zu den Quell-Repositorys für ein Projekt zu navigieren

1. Navigieren Sie zu Ihrem Projekt und führen Sie einen der folgenden Schritte aus:

- Wählen Sie auf der Übersichtsseite Ihres Projekts das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen.
  - Wählen Sie im Navigationsbereich Code und dann Quell-Repositorys aus. Wählen Sie unter Quell-Repositorys den Namen des Repositorys aus der Liste aus. Sie können die Liste der Repositorys filtern, indem Sie einen Teil des Repository-Namens in die Filterleiste eingeben.
2. Sehen Sie sich auf der Startseite des Repositorys den Inhalt des Repositorys und Informationen zu den zugehörigen Ressourcen an, z. B. die Anzahl der Pull-Requests und Workflows. Standardmäßig werden die Inhalte für den Standard-Branch angezeigt. Sie können die Ansicht ändern, indem Sie einen anderen Zweig aus der Dropdownliste auswählen.

## Schritt 5: Erstellen Sie eine Entwicklungsumgebung mit einem Testzweig und nehmen Sie eine schnelle Codeänderung vor

Sie können schnell am Code in Ihrem Quell-Repository arbeiten, indem Sie eine Entwicklungsumgebung erstellen. Für dieses Tutorial gehen wir davon aus, dass Sie:

- Erstellen Sie eine AWS Cloud9 Entwicklungsumgebung.
- Wählen Sie beim Erstellen der Entwicklungsumgebung die Option, in einem neuen Zweig des Hauptzweigs zu arbeiten.
- Verwenden Sie den Namen `test` für diesen neuen Zweig.

In einem späteren Schritt werden Sie die Entwicklungsumgebung verwenden, um eine Codeänderung vorzunehmen und eine Pull-Anfrage zu erstellen.

Um eine Entwicklungsumgebung mit einem neuen Branch zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung erstellen möchten.
3. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code, dann Quell-Repositorys und dann das Repository auswählen, für das Sie eine Entwicklungsumgebung erstellen möchten.
4. Wählen Sie auf der Repository-Startseite die Option Create Dev Environment aus.
5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).

6. Wählen Sie das zu klonende Repository aus, wählen Sie In neuem Zweig arbeiten aus, geben Sie einen Branch-Namen in das Feld Branch-Name ein und wählen Sie im Drop-down-Menü Branch erstellen einen Branch aus, aus dem der neue Branch erstellt werden soll.
7. Fügen Sie optional einen Alias für die Entwicklungsumgebung hinzu.
8. Wählen Sie optional die Schaltfläche „Konfiguration der Entwicklungsumgebung bearbeiten“, um die Rechen-, Speicher- oder Timeout-Konfiguration der Entwicklungsumgebung zu bearbeiten.
9. Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, wird in der Statusspalte Entwicklungsumgebung der Wert Wird gestartet angezeigt, und in der Statusspalte wird nach der Erstellung der Entwicklungsumgebung die Meldung Wird ausgeführt angezeigt. Es öffnet sich ein neuer Tab mit Ihrer Entwicklungsumgebung in der IDE Ihrer Wahl. Sie können Code bearbeiten und Ihre Änderungen übernehmen und pushen.

In diesem Abschnitt arbeiten Sie mit Ihrer generierten Beispielanwendung, CodeCatalyst indem Sie Änderungen am Code mit Pull-Requests vornehmen, die AWS-Konto beim Zusammenführen des Pull-Requests automatisch erstellt und für Ressourcen im verbundenen Netzwerk bereitgestellt werden.

Um eine Änderung an Ihrer `src/mysfit_data.json` Datei vorzunehmen

1. Navigieren Sie zu Ihrem Projekt Dev Environment. Erweitern Sie AWS Cloud9 darin das seitliche Navigationsmenü, um die Dateien zu durchsuchen. Erweitern Sie `mysfitssrc`, und öffnen `Siesrc/mysfit_data.json`.
2. Ändern Sie in der Datei den Wert für das "Age": Feld von 6 auf 12. Ihre Zeile sollte wie folgt aussehen:

```
{
  "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. Speichern Sie die Datei.
4. Wechseln Sie mit dem **cd /projects/mysfits** Befehl zum Mysfits-Repository.
5. Füge deine Änderungen mit den Befehlen `git add`, `git commit` und `git push` hinzu, bestätige sie und übertrage sie.

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

## Schritt 6: Sehen Sie sich den Workflow an, der die moderne Anwendung erstellt

CodeCatalyst Generiert nach der Erstellung des modernen Anwendungsprojekts in Ihrem Namen mehrere Ressourcen, einschließlich eines Workflows. Ein Workflow ist ein automatisiertes Verfahren, das in einer `.yaml`-Datei definiert ist und beschreibt, wie Sie Ihren Code erstellen, testen und bereitstellen.

In diesem Tutorial haben Sie einen Workflow CodeCatalyst erstellt und ihn automatisch gestartet, als Sie Ihr Projekt erstellt haben. (Der Workflow läuft möglicherweise noch, je nachdem, wie lange es her ist, dass Sie Ihr Projekt erstellt haben.) Verwenden Sie die folgenden Verfahren, um den Fortschritt des Workflows zu überprüfen, die generierten Protokolle und Testberichte zu überprüfen und schließlich zur URL der bereitgestellten Anwendung zu navigieren.

Um den Fortschritt des Workflows zu überprüfen

1. Wählen Sie in der CodeCatalyst Konsole im Navigationsbereich CI/CD und dann Workflows aus.

Eine Liste von Workflows wird angezeigt. Dies sind die Workflows, die der CodeCatalyst Blueprint generiert und gestartet hat, als Sie Ihr Projekt erstellt haben.

2. Beachten Sie die Liste der Workflows. Sie sollten vier sehen:

- Die beiden Workflows oben entsprechen dem `test` Zweig, den Sie zuvor erstellt haben [Schritt 5: Erstellen Sie eine Entwicklungsumgebung mit einem Testzweig und nehmen Sie eine schnelle Codeänderung vor](#). Diese Workflows sind Klone der Workflows in der `main` Filiale. Der `ApplicationDeploymentPipeline` ist nicht aktiv, da er für die Verwendung mit der `main` Filiale konfiguriert ist. Der `OnPullRequestWorkflow` wurde nicht ausgeführt, da keine Pull-Anfrage gestellt wurde.
- Die beiden Workflows unten entsprechen dem `main` Branch, der erstellt wurde, als Sie den Blueprint zuvor ausgeführt haben. Der `ApplicationDeploymentPipelineWorkflow` ist aktiv und die Ausführung läuft (oder ist abgeschlossen).

#### Note

Wenn die `ApplicationDeploymentPipeline`-Ausführung mit einem Build `@cdk_bootstrap` oder einem `DeployBackendFehler` fehlschlägt, liegt das möglicherweise daran, dass Sie zuvor die moderne dreistufige Webanwendung ausgeführt haben und dabei alte Ressourcen zurückgelassen haben, die zu einem Konflikt mit dem aktuellen Blueprint führen. Sie müssen diese alten Ressourcen löschen und dann den Workflow erneut ausführen. Weitere Informationen finden Sie unter [Bereinigen von -Ressourcen](#).

3. Wählen Sie unten den `ApplicationDeploymentPipelineWorkflow` aus, der dem `main` Zweig zugeordnet ist. Dieser Workflow wurde unter Verwendung des Quellcodes für den `main` Zweig ausgeführt.

Ein Workflow-Diagramm wird angezeigt. Das Diagramm zeigt mehrere Blöcke, von denen jeder eine Aufgabe oder Aktion darstellt. Die meisten Aktionen sind vertikal angeordnet, wobei die Aktionen oben vor den Aktionen unten ausgeführt werden. Aktionen, die nebeneinander angeordnet sind, laufen parallel ab. Aktionen, die zusammen gruppiert sind, müssen alle erfolgreich ausgeführt werden, bevor die Aktion unter ihnen gestartet werden kann.



Die Hauptblöcke sind:

- `WorkflowSource`— Dieser Block steht für Ihr Quell-Repository. Er zeigt unter anderem den Namen des Quell-Repositorys (`mysfits`) und den Commit, der den Workflow-Lauf automatisch gestartet hat. CodeCatalyst hat diesen Commit generiert, als Sie Ihr Projekt erstellt haben.
- `Build` — Dieser Block stellt eine Gruppierung von zwei Aktionen dar, die beide erfolgreich abgeschlossen werden müssen, damit die nächste Aktion gestartet werden kann.



- **DeployBackend**— Dieser Block stellt eine Aktion dar, die die Backend-Komponenten der Anwendung in der Cloud bereitstellt. AWS
  - **Tests** — Dieser Block stellt eine Gruppierung von zwei Testaktionen dar, die beide erfolgreich abgeschlossen werden müssen, damit die nächste Aktion gestartet werden kann.
  - **DeployFrontend**— Dieser Block stellt eine Aktion dar, die die Frontend-Komponenten der Anwendung in der Cloud bereitstellt. AWS
4. Wählen Sie die Registerkarte Definition (ganz oben). Die [Workflow-Definitionsdatei](#) wird auf der rechten Seite angezeigt. Die Datei enthält die folgenden wichtigen Abschnitte:
    - Ein **Triggers** Abschnitt oben. In diesem Abschnitt wird angegeben, dass der Workflow immer dann beginnen muss, wenn Code in den `main` Branch des Quell-Repositorys übertragen wird. Durch Pushes in andere Zweige (z. B. `test`) wird dieser Workflow nicht gestartet. Der Workflow wird mithilfe der Dateien in der `main` Filiale ausgeführt.
    - Ein **Actions** Abschnitt, unter `Triggers`. In diesem Abschnitt werden die Aktionen definiert, die Sie im Workflow-Diagramm sehen.
  5. Wählen Sie die Registerkarte Neuester Status (oben) und wählen Sie eine beliebige Aktion im Workflow-Diagramm aus.
  6. Wählen Sie auf der rechten Seite die Registerkarte Konfiguration aus, um die Konfigurationseinstellungen zu sehen, die von der Aktion während der letzten Ausführung verwendet wurden. Jede Konfigurationseinstellung hat eine passende Eigenschaft in der Workflow-Definitionsdatei.
  7. Lassen Sie die Konsole geöffnet und fahren Sie mit dem nächsten Verfahren fort.

Um die Build-Logs und Testberichte zu überprüfen

1. Wählen Sie die Registerkarte Neuester Status.
2. Wählen Sie im Workflow-Diagramm die `DeployFrontend`Aktion aus.
3. Warten Sie, bis die Aktion abgeschlossen ist. Achten Sie darauf, dass das Symbol „In Bearbeitung“ () zu einem Symbol für „Erfolg“ () wechselt.
4. Wählen Sie die Aktion `build_backend`.

5. Wählen Sie die Registerkarte Protokolle und erweitern Sie einige Abschnitte, um die Protokollnachrichten für diese Schritte anzuzeigen. Sie können Meldungen sehen, die sich auf das Backend-Setup beziehen.
6. Wählen Sie die Registerkarte Berichte und dann den `backend-coverage.xml` Bericht aus. CodeCatalyst zeigt den zugehörigen Bericht an. Der Bericht zeigt die durchgeführten Codeabdeckungstests und gibt den Anteil der Codezeilen an, die durch Tests erfolgreich validiert wurden, z. B. 80%.

Weitere Informationen zu Testberichten finden Sie unter [Testen mit Workflows in CodeCatalyst](#).


 Tip

Sie können Ihre Testberichte auch anzeigen, indem Sie im Navigationsbereich Berichte auswählen.

7. Lassen Sie die CodeCatalyst Konsole geöffnet und fahren Sie mit dem nächsten Verfahren fort.

Um zu bestätigen, dass die moderne Anwendung erfolgreich bereitgestellt wurde

1. Kehren Sie zum `ApplicationDeploymentPipelineWorkflow` zurück und wählen Sie den Link **Run-String** der letzten Ausführung aus.
2. Suchen Sie im Workflow-Diagramm nach der `DeployFrontendAktion` und wählen Sie den Link `App anzeigen` aus. Die Mysfit-Website wird angezeigt.

 Note

Wenn Sie den Link `App anzeigen` in der `DeployFrontendAktion` nicht sehen, stellen Sie sicher, dass Sie den Link `Run-ID` ausgewählt haben.

3. Suchen Sie nach dem Pegasus Mysfit namens `Twilight Glitter`. Notieren Sie sich den Wert für das Alter. Ist es 6. Sie werden eine Codeänderung vornehmen, um das Alter zu aktualisieren.

## Schritt 7: Bitten Sie andere, Ihre Änderungen zu überprüfen

Da du nun Änderungen in einem Branch mit dem Namen `hasstest`, kannst du andere bitten, sie zu überprüfen, indem du eine Pull-Anfrage erstellst. Führen Sie die folgenden Schritte aus, um

eine Pull-Anfrage zu erstellen, um die Änderungen aus dem `test` Branch in den `main` Branch zusammenzuführen.

Um einen Pull-Request zu erstellen

1. Navigiere zu deinem Projekt.
2. Führen Sie eine der folgenden Aktionen aus:
  - Wählen Sie im Navigationsbereich Code, dann Pull Requests und dann Create Pull Request aus.
  - Wählen Sie auf der Repository-Startseite die Option Mehr und dann Pull-Request erstellen aus.
  - Wähle auf der Projektseite die Option Pull-Request erstellen aus.
3. Stellen Sie im Quell-Repository sicher, dass das angegebene Quell-Repository das ist, das den festgeschriebenen Code enthält. Diese Option wird nur angezeigt, wenn Sie den Pull-Request nicht von der Hauptseite des Repositories aus erstellt haben.
4. Wählen Sie im Zielzweig den Zweig aus, in den der Code nach der Überprüfung zusammengeführt werden soll.
5. Wählen Sie im Quellzweig den Zweig aus, der den festgeschriebenen Code enthält.
6. Geben Sie im Feld Titel der Pull-Anfrage einen Titel ein, der anderen Benutzern hilft zu verstehen, was überprüft werden muss und warum.
7. (Optional) Geben Sie unter Beschreibung des Pull-Requests Informationen wie einen Link zu Problemen oder eine Beschreibung Ihrer Änderungen ein.

 Tip

Du kannst Beschreibung für mich schreiben wählen, damit CodeCatalyst automatisch eine Beschreibung der in der Pull-Anfrage enthaltenen Änderungen generiert wird. Sie können Änderungen an der automatisch generierten Beschreibung vornehmen, nachdem Sie sie dem Pull Request hinzugefügt haben.

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).


8. (Optional) Wählen Sie unter Probleme die Option Probleme verknüpfen aus und wählen Sie dann entweder ein Problem aus der Liste aus oder geben Sie dessen ID ein. Um die Verknüpfung eines Problems aufzuheben, wählen Sie das Symbol „Verknüpfung aufheben“.

9. (Optional) Wählen Sie unter Erforderliche Prüfer die Option Erforderliche Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Erforderliche Prüfer müssen die Änderungen genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.

 Note

Sie können einen Prüfer nicht gleichzeitig als erforderlichen Prüfer und als optionalen Prüfer hinzufügen. Sie können sich nicht selbst als Prüfer hinzufügen.

10. (Optional) Wählen Sie unter Optionale Prüfer die Option Optionale Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Optionale Prüfer müssen die Änderungen nicht als Anforderung genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.
11. Überprüfe die Unterschiede zwischen den Branches. Der Unterschied, der in einem Pull Request angezeigt wird, besteht in den Änderungen zwischen der Revision im Quell-Branch und der Merge-Base, bei der es sich um den Head-Commit des Ziel-Branches zum Zeitpunkt der Erstellung des Pull-Requests handelt. Wenn keine Änderungen angezeigt werden, sind die Branches möglicherweise identisch, oder Sie haben denselben Branch sowohl für die Quelle als auch für das Ziel ausgewählt.
12. Wenn Sie überzeugt sind, dass der Pull Request den Code und die Änderungen enthält, die Sie überprüfen möchten, wählen Sie Create aus.

 Note

Nachdem Sie den Pull Request erstellt haben, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @-Zeichen gefolgt vom Namen der Datei verwendest.

Wenn Sie den Pull Request erstellen, verwendet der OnPullRequestWorkflow zunächst die Quelldateien im test Branch. Während Ihre Prüfer Ihre Codeänderung genehmigen, können Sie die Ergebnisse beobachten, indem Sie den Workflow auswählen und sich die Testausgabe ansehen.

Nachdem Sie die Änderung überprüft haben, können Sie den Code zusammenführen. Wenn Sie den Code mit dem Standardzweig zusammenführen, wird automatisch der Workflow gestartet, der Ihre Änderungen erstellt und bereitstellt.

Um eine Pull-Anfrage von der CodeCatalyst Konsole aus zusammenzuführen

1. Navigieren Sie zu Ihrem modernen Anwendungsprojekt.
2. Wählen Sie auf der Projektseite unter Open Pull Requests den Pull Request aus, den Sie zusammenführen möchten. Wenn du den Pull Request nicht siehst, wähle Alle anzeigen und wähle ihn dann aus der Liste aus. Wählen Sie Merge (Zusammenführen).
3. Wählen Sie aus den verfügbaren Merge-Strategien für den Pull Request aus. Aktivieren oder deaktivieren Sie optional die Option, den Quell-Branch nach dem Zusammenführen des Pull-Requests zu löschen, und wählen Sie dann Merge.

#### Note

Wenn die Schaltfläche Zusammenführen nicht aktiv ist oder die Bezeichnung Nicht zusammenführbar angezeigt wird, haben entweder ein oder mehrere erforderliche Reviewer den Pull Request noch nicht genehmigt, oder der Pull Request kann nicht in der Konsole zusammengeführt werden. CodeCatalyst Ein Prüfer, der eine Pull-Anfrage nicht genehmigt hat, wird in der Übersicht im Bereich Pull-Request-Details durch ein Uhrensymbol gekennzeichnet. Wenn alle erforderlichen Prüfer den Pull Request genehmigt haben, die Schaltfläche Zusammenführen aber immer noch nicht aktiv ist, liegt möglicherweise ein Merge-Konflikt vor. Sie können Mergekonflikte für den Ziel-Branch in der CodeCatalyst Konsole lösen und dann den Pull Request zusammenführen, oder Sie können Konflikte lösen und lokal zusammenführen und dann den Commit, der den Merge enthält, per Push übertragen. CodeCatalyst Weitere Informationen finden Sie unter [Eine Pull-Anfrage zusammenführen \(Git\)](#) und in Ihrer Git-Dokumentation.

Sobald du die Änderungen aus dem Branch mit dem `test` Branch zusammengeführt hast, startet die Änderung automatisch den `ApplicationDeploymentPipelineWorkflow`, der deine Änderung erstellt und bereitstellt. **main**

Um zu sehen, wie der zusammengeführte Commit durch den `ApplicationDeploymentPipelineWorkflow` läuft

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.

2. Erweitern Sie unter Workflows in den ApplicationDeploymentPipelineBereich Letzte Ausführungen. Sie können die Workflow-Ausführung sehen, die durch den Merge-Commit gestartet wurde. Wählen Sie diese Option optional aus, um den Fortschritt der Ausführung zu verfolgen.
3. Wenn der Lauf abgeschlossen ist, laden Sie die URL, die Sie zuvor besucht haben, erneut. Sehen Sie sich den Pegasus an, um zu überprüfen, ob sich das Alter geändert hat.



## Schritt 8: Schließen Sie das Problem

Wenn ein Problem behoben ist, kann es auf der CodeCatalyst Konsole geschlossen werden.

Um ein Problem für ein Projekt zu schließen

1. Navigieren Sie zu Ihrem Projekt.
2. Wählen Sie im Navigationsbereich Issues aus.
3. Geben rag-and-drop Sie das Problem in die Spalte Fertig ein.

Weitere Informationen finden Sie unter [Probleme in CodeCatalyst](#).

## Bereinigen von -Ressourcen

Räumen Sie in Ihrer Umgebung AWS auf CodeCatalyst und entfernen Sie Spuren dieses Tutorials aus Ihrer Umgebung.

Sie können wählen, ob Sie das Projekt, das Sie für dieses Tutorial verwendet haben, weiterhin verwenden möchten, oder Sie können das Projekt und die zugehörigen Ressourcen löschen.

### Note

Wenn Sie dieses Projekt löschen, werden alle Repositorys, Probleme und Artefakte im Projekt für alle Mitglieder gelöscht.

So löschen Sie ein Projekt

1. Navigieren Sie zu Ihrem Projekt und wählen Sie dann Projekteinstellungen.
2. Wählen Sie die Registerkarte Allgemein.
3. Wählen Sie unter dem Projektnamen die Option Projekt löschen aus.

Um Ressourcen in AWS CloudFormation und Amazon S3 zu löschen

1. Melden Sie sich AWS Management Console mit demselben Konto an, das Sie Ihrem CodeCatalyst Bereich hinzugefügt haben.
2. Gehe zum AWS CloudFormationService.
3. Lösche den ***Mysfits-String-Stapel***.
4. ***Löschen Sie den Development-Mysfits-String-Stapel.***
5. Wählen Sie den CDKToolkit-Stack aus (löschen Sie ihn aber nicht). Wählen Sie die Registerkarte Resources (Ressourcen) aus. Wählen Sie den StagingBucketLink und löschen Sie den Bucket und den Bucket-Inhalt in Amazon S3.

### Note

Wenn Sie diesen Bucket nicht manuell löschen, wird möglicherweise ein Fehler angezeigt, wenn Sie den Blueprint für moderne dreistufige Webanwendungen erneut ausführen.

## 6. (Optional) Löschen Sie den CDKToolkit-Stack.

## Referenz

Der Blueprint für moderne dreistufige Webanwendungen stellt Ressourcen in Ihrem CodeCatalyst Bereich und in Ihrem AWS Konto in der Cloud bereit. AWS Diese Ressourcen sind:

- In deinem CodeCatalyst Raum:
  - Ein CodeCatalyst Projekt, das die folgenden Ressourcen umfasst:
    - Ein [Quell-Repository](#) — Dieses Repository enthält Beispielcode für eine „Mysfits“ - Webanwendung.
    - Ein [Workflow](#) — Dieser Workflow erstellt den Mysfits-Anwendungscode und stellt ihn bereit, wenn eine Änderung am Standardzweig vorgenommen wird
    - Ein [Forum und ein Backlog für Probleme](#) — Dieses Board und das Backlog können zur Planung und Nachverfolgung der Arbeit verwendet werden.
    - Eine [Suite für Testberichte](#) — Diese Suite enthält automatisierte Berichte, die im Beispielcode enthalten sind.
  - Im zugehörigen AWS-Konto:
    - Ein CDKToolkit-Stack — Dieser Stack stellt die folgenden Ressourcen bereit:
      - Ein Amazon S3 S3-Staging-Bucket, eine Bucket-Richtlinie und der AWS KMS Schlüssel, mit dem der Bucket verschlüsselt wurde.
      - Eine IAM-Bereitstellungsrolle für die Bereitstellungsaktion.
      - AWS IAM-Rollen und -Richtlinien zur Unterstützung der Ressourcen im Stack.

### Note

Das CDKToolkit wird nicht für jede Bereitstellung heruntergeladen und neu erstellt. Dies ist ein Stack, der in jedem Konto initiiert wird, um das zu unterstützen. AWS CDK

- Ein ***BackendDevelopment-Mysfits-String-Stack*** — Dieser Stack stellt die folgenden Backend-Ressourcen bereit:
  - Ein Amazon API Gateway Gateway-Endpunkt.
  - AWS IAM-Rollen und -Richtlinien zur Unterstützung der Ressourcen im Stack.



- Eine AWS Lambda Funktion und eine Ebene bilden die serverlose Rechenplattform für moderne Anwendungen.
- Eine IAM-Richtlinie und -Rolle für die Bucket-Bereitstellung und die Lambda-Funktion.
- Ein ***Mysfits-String-Stack*** — Dieser Stack stellt die Frontend-Anwendung bereit. AWS Amplify

Weitere Informationen finden Sie auch unter

Weitere Informationen zu den AWS Diensten, bei denen Ressourcen im Rahmen dieses Tutorials erstellt werden, finden Sie im Folgenden:

- Amazon S3 — Ein Service zum Speichern Ihrer Frontend-Ressourcen auf einem Objektspeicherservice, der branchenführende Skalierbarkeit, hohe Datenverfügbarkeit, Sicherheit und Leistung bietet. Weitere Informationen finden Sie im [Amazon S3 S3-Benutzerhandbuch](#).
- Amazon API Gateway — Ein Service für die Erstellung, Veröffentlichung, Wartung, Überwachung und Sicherung von REST, HTTP und WebSocket APIs in jeder Größenordnung. Weitere Informationen finden Sie im [API Gateway Developer Guide](#).
- Amplify — Ein Service zum Hosten Ihrer Frontend-Anwendung. Weitere Informationen finden Sie im [AWS Amplify Hosting-Benutzerhandbuch](#).
- AWS Cloud Development Kit (AWS CDK)— Ein Framework für die Definition der Cloud-Infrastruktur im Code und deren Bereitstellung. AWS CloudFormation AWS CDK Dazu gehört das AWS CDK Toolkit, ein Befehlszeilentool für die Interaktion mit AWS CDK Apps und Stacks. Weitere Informationen finden Sie im [AWS Cloud Development Kit \(AWS CDK\) -Entwicklerhandbuch](#).
- Amazon DynamoDB — Ein vollständig verwalteter NoSQL-Datenbankservice zum Speichern von Daten. Weitere Informationen finden Sie im [Amazon DynamoDB DynamoDB-Entwicklerhandbuch](#).
- AWS Lambda— Ein Service, mit dem Sie Ihren Code in einer hochverfügbaren Recheninfrastruktur aufrufen können, ohne Server bereitzustellen oder zu verwalten. Weitere Informationen finden Sie im [AWS Lambda -Entwicklerhandbuch](#).
- AWS IAM — Ein Dienst zur sicheren Steuerung des Zugriffs auf AWS und die zugehörigen Ressourcen. Weitere Informationen finden Sie im [IAM-Benutzerhandbuch](#).

# Tutorial: Mit einem leeren Projekt beginnen und Ressourcen manuell hinzufügen

Sie können ein leeres Projekt ohne vordefinierte Ressourcen erstellen, indem Sie beim Erstellen des Projekts den Blueprint Leeres Projekt auswählen. Nachdem Sie ein leeres Projekt erstellt haben, können Sie diesem entsprechend Ihren Projektanforderungen Ressourcen erstellen und hinzufügen. Da Projekte, die ohne Blueprint erstellt wurden, bei der Erstellung leer sind, erfordert diese Option mehr Wissen über das Erstellen und Konfigurieren von CodeCatalyst Ressourcen, um loszulegen.

## Themen

- [Voraussetzungen](#)
- [Erstellen Sie ein leeres Projekt](#)
- [Erstellen Sie ein Quell-Repository](#)
- [Erstellen Sie einen Workflow zum Erstellen, Testen und Bereitstellen einer Codeänderung](#)
- [Laden Sie jemanden zu Ihrem Projekt ein](#)
- [Erstellen Sie Probleme, an denen Sie gemeinsam arbeiten können, und verfolgen Sie die Arbeit](#)

## Voraussetzungen

Um ein leeres Projekt zu erstellen, muss Ihnen die Rolle Space-Administrator oder Poweruser zugewiesen sein. Wenn Sie sich zum ersten Mal anmelden CodeCatalyst, finden Sie weitere Informationen unter [Einrichten CodeCatalyst](#).

## Erstellen Sie ein leeres Projekt

Das Erstellen eines Projekts ist der erste Schritt, um zusammenarbeiten zu können. Wenn Sie Ihre eigenen Ressourcen wie Quell-Repositorys und Workflows erstellen möchten, können Sie mit einem leeren Projekt beginnen.

Um ein leeres Projekt zu erstellen

1. Navigieren Sie zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
2. Wählen Sie im Space-Dashboard die Option Projekt erstellen aus.
3. Wählen Sie „Von vorne beginnen“.
4. Geben Sie unter Geben Sie Ihrem Projekt einen Namen ein den Namen ein, den Sie Ihrem Projekt zuweisen möchten. Der Name muss in Ihrem Bereich eindeutig sein.

5. Wählen Sie **Create project** (Projekt erstellen) aus.

Jetzt, da Sie ein leeres Projekt haben, besteht der nächste Schritt darin, ein Quell-Repository zu erstellen.

## Erstellen Sie ein Quell-Repository

Erstellen Sie ein Quell-Repository, um den Code Ihres Projekts zu speichern und gemeinsam daran zu arbeiten. Projektmitglieder können dieses Repository auf ihre lokalen Computer klonen, um an Code zu arbeiten. Alternativ können Sie sich dafür entscheiden, ein Repository zu verknüpfen, das in einem unterstützten Dienst gehostet wird. Dies wird in diesem Tutorial jedoch nicht behandelt. Weitere Informationen finden Sie unter [Ein Quell-Repository verlinken](#).

Um ein Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt.
3. Wählen Sie im Navigationsbereich **Code** und dann **Quell-Repositories** aus.
4. Wählen Sie **Repository hinzufügen** und anschließend **Repository erstellen** aus.
5. Geben Sie im Feld **Repository-Name** einen Namen für das Repository ein. In diesem Handbuch verwenden wir einen anderen Namen *codecatalyst-source-repository*, aber Sie können einen anderen Namen wählen. Repository-Namen müssen innerhalb eines Projekts eindeutig sein. Weitere Hinweise zu den Anforderungen für Repository-Namen finden Sie unter [Kontingente für Quell-Repositories in CodeCatalyst](#).
6. (Optional) Fügen Sie unter **Beschreibung** eine Beschreibung für das Repository hinzu, damit andere Benutzer im Projekt verstehen, wofür das Repository verwendet wird.
7. (Optional) Fügen Sie eine `.gitignore` Datei für den Codetyp hinzu, den Sie pushen möchten.
8. Wählen Sie **Erstellen**.

### Note

CodeCatalyst fügt Ihrem Repository eine `README.md` Datei hinzu, wenn Sie sie erstellen. CodeCatalyst erstellt außerdem einen ersten Commit für das Repository in einem Standard-Branch namens `main`. Sie können die Datei `README.md` bearbeiten oder löschen, aber Sie können den Standard-Branch nicht ändern oder löschen.

Sie können Ihrem Repository schnell Code hinzufügen, indem Sie eine Entwicklungsumgebung erstellen. Für dieses Tutorial empfehlen wir Ihnen, eine Entwicklungsumgebung mithilfe AWS Cloud9 von zu erstellen und bei der Erstellung der Entwicklungsumgebung die Option zum Erstellen eines Zweigs aus dem Hauptzweig auszuwählen. Wir verwenden den Namen **test** für diesen Zweig, aber Sie können auch einen anderen Zweignamen eingeben, wenn Sie dies bevorzugen.

Um eine Entwicklungsumgebung mit einem neuen Zweig zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung erstellen möchten.
3. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code, dann Quell-Repositorys und dann das Repository auswählen, für das Sie eine Entwicklungsumgebung erstellen möchten.
4. Wählen Sie auf der Repository-Startseite die Option Create Dev Environment aus.
5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
6. Wählen Sie das zu klonende Repository aus, wählen Sie In neuem Zweig arbeiten aus, geben Sie einen Branch-Namen in das Feld Branch-Name ein und wählen Sie im Drop-down-Menü Branch erstellen einen Branch aus, aus dem der neue Branch erstellt werden soll.
7. Fügen Sie optional einen Alias für die Entwicklungsumgebung hinzu.
8. Wählen Sie optional die Schaltfläche „Konfiguration der Entwicklungsumgebung bearbeiten“, um die Rechen-, Speicher- oder Timeout-Konfiguration der Entwicklungsumgebung zu bearbeiten.
9. Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, wird in der Spalte Entwicklungsumgebung der Wert Wird gestartet angezeigt und in der Spalte wird nach Erstellung der Entwicklungsumgebung ausgeführt angezeigt. Es öffnet sich ein neuer Tab mit Ihrer Entwicklungsumgebung in der IDE Ihrer Wahl. Sie können Code bearbeiten und Ihre Änderungen übernehmen und pushen.

## Erstellen Sie einen Workflow zum Erstellen, Testen und Bereitstellen einer Codeänderung

In CodeCatalyst organisieren Sie das Erstellen, Testen und Bereitstellen Ihrer Anwendungen oder Dienste in Workflows. Workflows bestehen aus Aktionen und können so konfiguriert werden, dass sie automatisch ausgeführt werden, nachdem bestimmte Ereignisse im Quell-Repository eingetreten

sind, z. B. Code-Pushs oder das Öffnen oder Aktualisieren einer Pull-Anfrage. Weitere Informationen zu Workflows finden Sie unter [Erstellen, Testen und Bereitstellen mit Workflows in CodeCatalyst](#).

Folgen Sie den Anweisungen unter [Erste Schritte mit Workflows in CodeCatalyst](#), um Ihren ersten Workflow zu erstellen.

## Laden Sie jemanden zu Ihrem Projekt ein

Nachdem Sie Ihr benutzerdefiniertes Projekt eingerichtet haben, laden Sie andere ein, mit Ihnen zu arbeiten.

Um jemanden zu Ihrem Projekt einzuladen

1. Navigieren Sie zu dem Projekt, zu dem Sie Benutzer einladen möchten.
2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Wählen Sie auf der Registerkarte Mitglieder die Option Einladen aus.
4. Geben Sie die E-Mail-Adressen der Personen ein, die Sie als Benutzer für Ihr Projekt einladen möchten. Sie können mehrere E-Mail-Adressen eingeben, die durch ein Leerzeichen oder ein Komma getrennt sind. Sie können auch Mitglieder Ihres Bereichs auswählen, die nicht Mitglieder des Projekts sind.
5. Wählen Sie die Rolle für den Benutzer aus.

Wenn Sie mit dem Hinzufügen von Benutzern fertig sind, wählen Sie Einladen.

## Erstellen Sie Probleme, an denen Sie gemeinsam arbeiten können, und verfolgen Sie die Arbeit


CodeCatalyst hilft dir dabei, Funktionen, Aufgaben, Bugs und alle anderen Arbeiten im Zusammenhang mit Problemen im Zusammenhang mit deinem Projekt nachzuverfolgen. Sie können Probleme erstellen, um die benötigte Arbeit und Ideen nachzuverfolgen. Wenn du ein Problem erstellst, wird es standardmäßig zu deinem Backlog hinzugefügt. Du kannst Probleme in ein Board verschieben, in dem du die laufenden Arbeiten nachverfolgen kannst. Du kannst ein Problem auch einem bestimmten Projektmitglied zuweisen.

Um ein Problem für ein Projekt zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

Stellen Sie sicher, dass Sie in dem Projekt navigieren, in dem Sie Probleme erstellen möchten. Um alle Projekte anzuzeigen, wählen Sie im Navigationsbereich Amazon CodeCatalyst und bei Bedarf Alle Projekte anzeigen. Wählen Sie das Projekt aus, in dem Sie Probleme erstellen oder bearbeiten möchten.

2. Wählen Sie im Navigationsbereich Track und anschließend Backlog aus.
3. Wählen Sie „Problem erstellen“.
4. Geben Sie im Feld Titel der Ausgabe einen Namen für das Problem ein. Geben Sie optional eine Beschreibung des Problems an. Wählen Sie bei Bedarf den Status, die Priorität und die Schätzung für das Problem aus. Sie können das Problem auch einem Projektmitglied aus der Liste der Projektmitglieder zuweisen.

 Tip

Sie können Amazon Q ein Problem zuweisen, damit Amazon Q versucht, das Problem zu lösen. Wenn der Versuch erfolgreich ist, wird eine Pull-Anfrage erstellt und der Status des Problems ändert sich zu Wird geprüft, sodass Sie den Code überprüfen und testen können. Weitere Informationen finden Sie unter [Tutorial: Verwenden CodeCatalyst generativer KI-Funktionen zur Beschleunigung Ihrer Entwicklungsarbeit](#).

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).

5. Wählen Sie Speichern.

Nachdem Sie Probleme erstellt haben, können Sie sie Projektmitgliedern zuweisen, sie einschätzen und sie auf einem Kanban-Board verfolgen. Weitere Informationen finden Sie unter [Probleme in CodeCatalyst](#).

## Tutorial: Verwenden CodeCatalyst generativer KI-Funktionen zur Beschleunigung Ihrer Entwicklungsarbeit

Die generativen KI-Funktionen in Amazon CodeCatalyst befinden sich in der Vorschauversion und können sich ändern. Sie sind nur in der Region USA West (Oregon) verfügbar. Der Zugriff auf generative KI-Features variiert je nach Stufe. Weitere Informationen finden Sie unter [-Preisgestaltung](#).

Wenn Sie ein Projekt und ein Quell-Repository in Amazon in einem Bereich haben, CodeCatalyst in dem generative KI-Funktionen aktiviert sind, können Sie diese Funktionen verwenden, um die Softwareentwicklung zu beschleunigen. Entwickler müssen häufig mehr Aufgaben erledigen als Zeit, um sie zu erledigen. Oft nehmen sie sich nicht die Zeit, ihren Teammitgliedern ihre Codeänderungen zu erklären, wenn sie Pull-Anfragen zur Überprüfung dieser Änderungen erstellen, und erwarten, dass andere Benutzer die Änderungen selbsterklärend finden. Ersteller und Prüfer von Pull-Anfragen haben auch keine Zeit, alle Kommentare zu einer Pull-Anfrage gründlich zu finden und zu lesen, insbesondere wenn die Pull-Anfrage mehrere Revisionen hat. CodeCatalyst enthält generative KI-Funktionen, mit denen beide Teammitglieder ihre Aufgaben schneller erledigen können, und erhöhen die Zeit, die sie haben, sich auf die wichtigsten Teile ihrer Arbeit zu konzentrieren.

### Note

Powered by Amazon Bedrock AWS :implements [automatisierte Missbrauchserkennung](#). Da die Funktionen „Beschreibung schreiben“ für mich, „Inhaltszusammenfassung erstellen“ und „Probleme an die Amazon-Q-Funktionsentwicklung zuweisen“ auf Amazon Bedrock basieren, können Benutzer die in Amazon Bedrock implementierten Kontrollen voll ausnutzen, um Sicherheit und den verantworteten Einsatz künstlicher Intelligenz (KI) durchzusetzen.

In diesem Tutorial erfahren Sie, wie Sie die generativen KI-Funktionen in verwenden, CodeCatalyst um Änderungen zwischen Verzweigungen beim Erstellen von Pull-Anforderungen zusammenzufassen und Kommentare zu einer Pull-Anforderung zusammenzufassen. Sie erfahren auch, wie Sie Probleme mit Ihren Ideen für einfache Codeänderungen oder Verbesserungen erstellen und sie Amazon Q zuweisen.

## Voraussetzungen

Um mit den CodeCatalyst Funktionen in diesem Tutorial arbeiten zu können, müssen Sie zuerst abgeschlossen haben und Zugriff auf die folgenden Ressourcen haben:

- Sie haben eine - AWS Entwickler-ID oder eine SSO-Identität (Single Sign-On) für die Anmeldung bei CodeCatalyst.
- Ihr Projekt befindet sich in einem Bereich, in dem generative KI-Funktionen aktiviert sind. Weitere Informationen finden Sie unter [Verwalten von generativen KI-Features](#).
- Sie haben die Rolle Contributor- oder Projektadministrator in einem Projekt in diesem Bereich.

- Für das Projekt ist mindestens ein Quell-Repository konfiguriert. Verknüpfte Repositories werden nicht unterstützt.
- Bei der Zuweisung von Problemen, damit eine erste Lösung mit generativer KI erstellt wird, kann das Projekt nicht mit der Erweiterung Jira Software konfiguriert werden. Die Erweiterung wird für diese Funktion nicht unterstützt.

Weitere Informationen dazu finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#), [Probleme in CodeCatalyst](#), [Erweiterungen in CodeCatalyst](#) und [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

Dieses Tutorial basiert auf einem Projekt, das mit der Vorlage für moderne dreistufige Webanwendungen mit Python erstellt wurde. Wenn Sie ein Projekt verwenden, das mit einem anderen Blueprint erstellt wurde, können Sie die Schritte trotzdem ausführen, aber einige Details variieren, z. B. Beispielcode und Sprache.

## Erstellen einer Zusammenfassung der Codeänderungen zwischen Verzweigungen beim Erstellen einer Pull-Anforderung

Eine Pull-Anforderung ist die primäre Methode, mit der Sie und andere Projektmitglieder Codeänderungen von einer Verzweigung zur anderen überprüfen, kommentieren und zusammenführen können. Sie können Pull-Anfragen verwenden, um Codeänderungen gemeinsam auf kleinere Änderungen oder Korrekturen, wichtige Funktionserweiterungen oder neue Versionen Ihrer veröffentlichten Software zu überprüfen. Die Zusammenfassung der Codeänderungen und die Absicht hinter den Änderungen als Teil der Beschreibung der Pull-Anforderung ist für andere hilfreich, die den Code überprüfen werden, und hilft auch dabei, die Änderungen am Code im Laufe der Zeit zu verstehen. Entwickler verlassen sich jedoch häufig auf ihren Code, um sich selbst zu erklären oder mehrdeutige Details anzugeben, anstatt ihre Änderungen mit ausreichenden Details zu beschreiben, damit die Prüfer verstehen, was sie überprüfen oder was die Absicht hinter den Änderungen im Code war.

Sie können die Funktion Beschreibung für mich schreiben verwenden, wenn Sie Pull-Anforderungen erstellen, damit Amazon Q eine Beschreibung der Änderungen erstellt, die in einer Pull-Anforderung enthalten sind. Wenn Sie diese Option wählen, analysiert Amazon Q die Unterschiede zwischen der Quellverzweigung, die die Codeänderungen enthält, und der Zielverzweigung, in der Sie diese Änderungen zusammenführen möchten. Anschließend wird eine Zusammenfassung dessen erstellt, was diese Änderungen sind, sowie seine beste Interpretation der Absicht und Wirkung dieser Änderungen.



Sie können diese Funktion mit jeder von Ihnen erstellten Pull-Anfrage ausprobieren, aber in diesem Tutorial testen wir sie, indem wir einige einfache Änderungen am Code vornehmen, der in einem Projekt enthalten ist, das in einer Python-basierten Vorlage für moderne dreistufige Webanwendungen erstellt wurde.

### Tip

Wenn Sie ein Projekt verwenden, das mit einem anderen Blueprint oder Ihrem eigenen Code erstellt wurde, können Sie trotzdem diesem Tutorial folgen, aber die Beispiele in diesem Tutorial stimmen nicht mit dem Code in Ihrem Projekt überein. Nehmen Sie anstelle des unten vorgeschlagenen Beispiels einfache Änderungen am Code Ihres Projekts in einer Verzweigung vor und erstellen Sie dann eine Pull-Anforderung, um das Feature zu testen, wie in den folgenden Schritten gezeigt.

Zunächst erstellen Sie eine Verzweigung im Quell-Repository. Anschließend nehmen Sie mithilfe des Texteditors in der Konsole eine Schnellcodeänderung an einer Datei in dieser Verzweigung vor. Anschließend erstellen Sie eine Pull-Anforderung und verwenden die Funktion Beschreibung für mich schreiben, um die von Ihnen vorgenommenen Änderungen zusammenzufassen.

So erstellen Sie eine Verzweigung (Konsole)

1. Navigieren Sie in der - CodeCatalyst Konsole zu dem Projekt, in dem sich Ihr Quell-Repository befindet.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Alternativ können Sie im Navigationsbereich Code und dann Quell-Repositorys auswählen.
3. Wählen Sie das Repository aus, in dem Sie einen Zweig erstellen möchten.
4. Wählen Sie auf der Übersichtsseite des Repositorys Weitere und dann Zweig erstellen aus.
5. Geben Sie einen Namen für die Verzweigung ein.
6. Wählen Sie einen Zweig aus, aus dem der Zweig erstellt werden soll, und wählen Sie dann Erstellen aus.

Sobald Sie eine Verzweigung haben, bearbeiten Sie eine Datei in dieser Verzweigung mit einer einfachen Änderung. In diesem Beispiel bearbeiten Sie die `-test_endpoint.py` Datei, um die Anzahl der Wiederholungen für Tests von auf **5 3** zu ändern.

**Tip**

Sie können auch eine Entwicklungsumgebung erstellen oder verwenden, um diese Codeänderung vorzunehmen. Weitere Informationen finden Sie unter [Erstellen einer Entwicklungsumgebung](#).

So bearbeiten Sie die **test\_endpoint.py** Datei in der Konsole

1. Wählen Sie auf der Übersichtsseite für das **mysfits** Quell-Repository das Drop-down-Menü für die Verzweigung und wählen Sie die Verzweigung aus, die Sie im vorherigen Verfahren erstellt haben.
2. Navigieren Sie unter Dateien zu der Datei, die Sie bearbeiten möchten. Um beispielsweise die `test_endpoint.py` Datei zu bearbeiten, erweitern Sie Tests, erweitern Sie Ganzzahl und wählen Sie dann `austest_endpoint.py`.
3. Wählen Sie Bearbeiten aus.
4. Ändern Sie in Zeile 7 die Häufigkeit, mit der alle Tests wiederholt werden:

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

auf:

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

5. Klicken Sie auf Commit und übernehmen Sie Ihre Änderungen in Ihrer Verzweigung.

Nachdem Sie nun eine Verzweigung mit einer Änderung haben, können Sie eine Pull-Anforderung erstellen.

Erstellen einer Pull-Anforderung mit einer Zusammenfassung der Änderungen

1. Wählen Sie auf der Übersichtsseite des Repositorys Weitere und dann Pull-Anforderung erstellen aus.
2. Wählen Sie unter Zielzweig den Zweig aus, in den der Code nach der Überprüfung zusammengeführt werden soll.

 Tip

Wählen Sie den Zweig aus, aus dem Sie Ihren Zweig im vorherigen Verfahren erstellt haben, um die einfachste Demonstration dieser Funktion zu erhalten. Wenn Sie beispielsweise Ihre Verzweigung aus der Standardverzweigung des Repositorys erstellt haben, wählen Sie diese Verzweigung als Zielverzweigung für Ihre Pull-Anforderung aus.

3. Wählen Sie unter Quellzweig den Zweig aus, der die Änderungen enthält, die Sie gerade an die `test_endpoint.py` Datei übergeben haben.
4. Geben Sie unter Titel der Pull-Anforderung einen Titel ein, der anderen Benutzern hilft zu verstehen, was überprüft werden muss und warum.
5. Wählen Sie unter Beschreibung der Pull-Anforderung die Option Beschreibung schreiben aus, damit eine Beschreibung der in der Pull-Anforderung enthaltenen Änderungen CodeCatalyst erstellt.
6. Eine Zusammenfassung der Änderungen wird angezeigt. Überprüfen Sie den vorgeschlagenen Text und wählen Sie Annehmen und zur Beschreibung hinzufügen aus.
7. Ändern Sie optional die Zusammenfassung, um die Änderungen, die Sie am Code vorgenommen haben, besser widerzuspiegeln. Sie können dieser Pull-Anforderung auch Prüfer hinzufügen oder Probleme verknüpfen. Wenn Sie alle gewünschten zusätzlichen Änderungen vorgenommen haben, wählen Sie Erstellen aus.

## Erstellen einer Zusammenfassung der Kommentare zu Codeänderungen in einer Pull-Anforderung

Wenn Benutzer eine Pull-Anforderung überprüfen, hinterlassen sie häufig mehrere Kommentare zu den Änderungen in dieser Pull-Anforderung. Wenn es viele Kommentare von vielen Prüfern gibt, kann es schwierig sein, im Feedback gemeinsame Themen auszuwählen oder sogar sicherzustellen, dass Sie alle Kommentare in allen Revisionen überprüft haben. Sie können die Funktion Kommentarzusammenfassung erstellen verwenden, damit Amazon Q alle Kommentare zu Codeänderungen in einer Pull-Anforderung analysiert und eine Zusammenfassung dieser Kommentare erstellt.

**Note**

Kommentarzusammenfassungen sind vorübergehend. Wenn Sie eine Pull-Anforderung aktualisieren, verschwindet die Zusammenfassung. Inhaltzusammenfassungen enthalten keine Kommentare zur gesamten Pull-Anforderung, sondern nur Kommentare zu Codeunterschieden in Revisionen der Pull-Anforderung.

So erstellen Sie eine Zusammenfassung der Kommentare in einer Pull-Anforderung

1. Navigieren Sie zu der Pull-Anforderung, die Sie im vorherigen Verfahren erstellt haben.

**Tip**

Wenn Sie möchten, können Sie jede offene Pull-Anfrage in Ihrem Projekt verwenden. Wählen Sie in der Navigationsleiste Code , wählen Sie Pull-Anforderungen und wählen Sie eine beliebige offene Pull-Anforderung aus.

2. Fügen Sie der Pull-Anforderung unter Änderungen einige Kommentare hinzu, wenn die Pull-Anforderung noch keine Kommentare enthält.
3. Wählen Sie unter Übersicht die Option Kommentarzusammenfassung erstellen aus. Wenn der Vorgang abgeschlossen ist, wird der Abschnitt Kommentarzusammenfassung erweitert.
4. Überprüfen Sie die Zusammenfassung der Kommentare zu Änderungen im Code in Revisionen der Pull-Anforderung und vergleichen Sie sie mit den Kommentaren in der Pull-Anforderung.

## Erstellen eines Problems und Zuweisen zu Amazon Q

Entwicklungsteams erstellen Probleme, um ihre Arbeit zu verfolgen und zu verwalten, aber manchmal bleibt ein Problem bestehen, da entweder nicht klar ist, wer daran arbeiten soll, das Problem eine Untersuchung eines bestimmten Teils der Codebasis erfordert oder andere dringende Arbeit zuerst durchgeführt werden muss. CodeCatalyst umfasst die Integration mit einem generativen KI-Assistenten namens Amazon Q, der ein Problem basierend auf seinem Titel und seiner Beschreibung analysieren kann. Wenn Sie das Problem Amazon Q zuweisen, versucht es, einen Lösungsentwurf zu erstellen, den Sie bewerten können. Auf diese Weise können Sie und Ihr Team Ihre Arbeit auf Probleme konzentrieren und optimieren, die Ihre Aufmerksamkeit erfordern, während Amazon Q an einer Lösung für Probleme arbeitet, die Sie nicht sofort lösen können.

 Tip

Amazon Q bietet bei einfachen und unkomplizierten Problemen die beste Leistung. Um optimale Ergebnisse zu erzielen, verwenden Sie Klarsprache, um klar zu erklären, was Sie tun möchten.

Wenn Sie Amazon Q ein Problem zuweisen, CodeCatalyst markiert das Problem als blockiert, bis Sie bestätigen, wie Amazon Q an dem Problem arbeiten soll. Sie müssen drei Fragen beantworten, bevor Sie fortfahren können:

- Ob Sie jeden Schritt bestätigen möchten oder ob Sie möchten, dass er ohne Feedback fortgesetzt wird. Wenn Sie sich dafür entscheiden, jeden Schritt zu bestätigen, können Sie Amazon Q mit Feedback zu dem Ansatz antworten, den es erstellt, damit es bei Bedarf seinen Ansatz wiederholen kann. Amazon Q kann auch Feedback-Benutzer bei jeder erstellten Pull-Anfrage überprüfen, wenn Sie diese Option wählen. Wenn Sie sich dafür entscheiden, die einzelnen Schritte nicht zu bestätigen, kann Amazon Q seine Arbeit schneller erledigen, überprüft jedoch kein Feedback, das Sie ihm im Problem oder in einer von ihm erstellten Pull-Anforderung geben.
- Gibt an, ob Sie zulassen möchten, dass Workflow-Dateien im Rahmen ihrer Arbeit aktualisiert werden. In Ihrem Projekt sind möglicherweise Workflows konfiguriert, um Ausführungen von Pull-Anforderungsereignissen zu starten. In diesem Fall kann jede von Amazon Q erstellte Pull-Anforderung, die das Erstellen oder Aktualisieren von Workflow-YAML beinhaltet, eine Ausführung dieser Workflows starten, die in der Pull-Anforderung enthalten sind. Als bewährte Methode sollten Sie Amazon Q nicht erlauben, an Workflow-Dateien zu arbeiten, es sei denn, Sie sind sicher, dass Ihr Projekt keine Workflows enthält, die diese Workflows automatisch ausführen, bevor Sie die erstellte Pull-Anforderung überprüfen und genehmigen.
- In welchem Quell-Repository es funktionieren soll. Selbst wenn Ihr Projekt über mehrere Quell-Repositorys verfügt, kann Amazon Q nur an Code in einem Quell-Repository arbeiten. Verknüpfte Repositorys werden nicht unterstützt.

Sobald Sie Ihre Auswahl getroffen und bestätigt haben, verschiebt Amazon Q das Problem in den Status In Bearbeitung, während versucht wird, festzustellen, was die Anforderung auf dem Titel des Problems und seiner Beschreibung sowie auf dem Code im angegebenen Repository basiert. Es wird einen angehefteten Kommentar erstellen, in dem es Updates zum Status seiner Arbeit bereitstellt. Nach der Überprüfung der Daten formuliert Amazon Q einen potenziellen Ansatz für eine Lösung. Amazon Q zeichnet seine Aktionen auf, indem es seinen angehefteten Kommentar

aktualisiert und in jeder Phase seinen Fortschritt zum Problem kommentiert. Im Gegensatz zu angehefteten Kommentaren und Antworten führt sie keine strikte chronologische Aufzeichnung seiner Arbeit. Stattdessen werden die relevantesten Informationen über seine Arbeit auf der obersten Ebene des angehefteten Kommentars abgelegt. Es wird versuchen, Code auf der Grundlage seines Ansatzes und seiner Analyse des Codes zu erstellen, der sich bereits im Repository befindet. Wenn es erfolgreich eine potenzielle Lösung generiert, erstellt es eine Verzweigung und Commit-Code für diese Verzweigung. Anschließend wird eine Pull-Anforderung erstellt, die diese Verzweigung mit der Standardverzweigung zusammenführt. Wenn Amazon Q seine Arbeit abgeschlossen hat, verschiebt es das Problem in In Überprüfung, sodass Sie und Ihr Team wissen, dass Code für die Bewertung bereit ist.

#### Note

Diese Funktion ist nur über Probleme verfügbar. Sie ist nicht verfügbar, wenn Sie Ihr Projekt für die Verwendung von J Bol mit der Jira-Softwareerweiterung konfiguriert haben. Wenn Sie das Layout Ihres Bords angepasst haben, ändert das Problem möglicherweise nicht den Status. Verwenden Sie diese Funktion nur für Projekte mit einem Standard-Board-Layout, um optimale Ergebnisse zu erzielen.

Sobald Sie Amazon Q ein Problem zugewiesen haben, können Sie den Titel oder die Beschreibung des Problems nicht mehr ändern oder ihn anderen zuweisen. Wenn Sie die Zuweisung von Amazon Q zum Problem aufheben, wird der aktuelle Schritt beendet und die Arbeit beendet. Sie kann die Arbeit nicht fortsetzen oder dem Problem neu zugewiesen werden, sobald sie nicht mehr zugewiesen wurde.

In diesem Teil des Tutorials erstellen Sie drei Probleme, die auf potenziellen Features für den Code basieren, der in Projekten enthalten ist, die mit dem Blueprint Modern für dreistufige Webanwendungen erstellt wurden: eine, um eine zu erstellen, eine, um eine neue mysfit-Kreatur hinzuzufügen, eine, um ein Sortier-Feature hinzuzufügen, und eine, um einen Workflow zu aktualisieren, um eine Verzweigung mit dem Namen einzuschließen **test**.

#### Note

Wenn Sie in einem Projekt mit anderem Code arbeiten, erstellen Sie Probleme mit Titeln und Beschreibungen, die sich auf diese Codebasis beziehen.

## So erstellen Sie ein Problem und lassen eine Lösung zur Bewertung generiert

1. Wählen Sie im Navigationsbereich Probleme und stellen Sie sicher, dass Sie sich in der Pinnenansicht befinden.
2. Wählen Sie Problem erstellen aus.
3. Geben Sie dem Problem einen Titel, der erklärt, was Sie in einfacher Sprache tun möchten. Geben Sie beispielsweise für dieses Problem den Titel ein **Create another mysfit named Quokkapus**. Geben Sie unter Beschreibung die folgenden Details ein:

```
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. (Optional) Fügen Sie ein Bild an, das als Miniaturansicht und Profilbild für mysfit für das Problem verwendet werden soll. Aktualisieren Sie in diesem Fall die Beschreibung, um Details darüber aufzunehmen, welche Bilder Sie verwenden möchten und warum. Sie können beispielsweise Folgendes zur Beschreibung hinzufügen: „Der mysfit erfordert, dass Image-Dateien auf der Website bereitgestellt werden. Fügen Sie diese Images dem Quell-Repository hinzu.“

 Note

Angefügte Images werden in diesem Tutorial nicht auf der Website bereitgestellt. Sie können die Bilder selbst zur Website hinzufügen und dann Kommentare für Amazon Q hinterlassen, um seinen Code so zu aktualisieren, dass er auf die Bilder verweist, die er verwenden soll, nachdem er eine Pull-Anforderung erstellt hat.

Überprüfen Sie die Beschreibung und stellen Sie sicher, dass sie alle Details enthält, die möglicherweise erforderlich sind, bevor Sie mit dem nächsten Schritt fortfahren.

5. Wählen Sie unter Zugewiesene die Option Amazon Q zuweisen aus.
6. Wählen Sie unter Quell-Repository das Quell-Repository aus, das den Projektcode enthält.
7. Führen Sie Amazon Q auffordern, nach jedem Schritt zu stoppen und auf die Überprüfung seiner Arbeitsauswahl auf den aktiven Zustand zu warten.


 Note

Wenn Sie die Option wählen, dass Amazon Q nach jedem Schritt beendet wird, können Sie das Problem kommentieren und die Möglichkeit haben, dass Amazon Q seinen Ansatz basierend auf Ihren Kommentaren bis zu dreimal ändert. Wenn Sie sich dafür entscheiden, Amazon Q nach jedem Schritt nicht stoppen zu lassen, damit Sie seine Arbeit überprüfen können, kann die Arbeit schneller fortgesetzt werden, da Amazon Q nicht auf Ihr Feedback wartet, Sie aber nicht in der Lage sind, die Richtung zu beeinflussen, die Amazon Q durch Verlassen von Kommentaren geht. Amazon Q antwortet auch nicht auf Kommentare, die in einer Pull-Anforderung verbleiben, wenn Sie diese Option wählen.

8. Belassen Sie die Auswahl Amazon Q erlauben, Workflow-Dateien zu ändern im inaktiven Status.
9. Wählen Sie Problem erstellen aus. Ihre Ansicht ändert sich am Issues-Board.
10. Wählen Sie Problem erstellen, um ein weiteres Problem zu erstellen, dieses Mal mit dem Titel **Change the `get_all_mysfits()` API to return `mysfits` sorted by the `Age` attribute**. Weisen Sie dieses Problem Amazon Q zu und erstellen Sie das Problem.
11. Wählen Sie Problem erstellen, um ein weiteres Problem zu erstellen, dieses Mal mit dem Titel **Update the `OnPullRequest` workflow to include a branch named `test` in its `triggers`**. Verknüpfen Sie optional mit dem Workflow in der Beschreibung. Weisen Sie




dieses Problem Amazon Q zu, stellen Sie jedoch sicher, dass die Auswahl Amazon Q erlauben, Workflow-Dateien zu ändern auf den aktiven Status gesetzt ist. Erstellen Sie das Problem, um zum Issues-Board zurückzukehren.

 Tip

Sie können nach Dateien suchen, einschließlich Workflow-Dateien, indem Sie das At-Symbol (@) und den Dateinamen eingeben.

Sobald Sie die Probleme erstellt und zugewiesen haben, werden die Probleme in In Bearbeitung verschoben. Amazon Q fügt Kommentare hinzu, die den Fortschritt innerhalb des Problems in einem angehefteten Kommentar verfolgen. Wenn es in der Lage ist, einen Ansatz für eine Lösung zu definieren, aktualisiert es die Beschreibung des Problems mit einem Hintergrundabschnitt, der die Analyse der Codebasis und einen Ansatzabschnitt enthält, der den vorgeschlagenen Ansatz zum Erstellen einer Lösung beschreibt. Wenn Amazon Q erfolgreich eine Lösung für das im Problem beschriebene Problem finden kann, erstellt es einen Zweig und Codeänderungen in diesem Zweig, die die vorgeschlagene Lösung implementieren. Sobald der Code fertig ist, erstellt er eine Pull-Anforderung, damit Sie die vorgeschlagenen Codeänderungen überprüfen können, fügt einen Link zu dieser Pull-Anforderung zu dem Problem hinzu und verschiebt das Problem in In Überprüfung.

 Important

Sie sollten Codeänderungen in einer Pull-Anforderung immer überprüfen, bevor Sie sie zusammenführen. Das Zusammenführen von Codeänderungen, die von Amazon Q vorgenommen wurden, kann sich wie alle anderen Codeänderungen negativ auf Ihre Codebasis und Ihren Infrastrukturcode auswirken, wenn der zusammengeführte Code nicht ordnungsgemäß überprüft wird und bei der Zusammenführung Fehler enthält.

So überprüfen Sie ein Problem und eine verknüpfte Pull-Anforderung, die von Amazon Q vorgenommene Änderungen enthält

1. Wählen Sie unter Probleme ein Problem aus, das Amazon Q zugewiesen ist und in Bearbeitung ist. Überprüfen Sie die Kommentare, um den Fortschritt des Bots zu überwachen. Falls vorhanden, überprüfen Sie den Hintergrund und nähern Sie sich den Datensätzen in der Beschreibung des Problems. Wählen Sie dann X aus, um den Problembereich zu schließen.

2. Wählen Sie jetzt ein Problem aus, das Amazon Q zugewiesen ist und sich in In Überprüfung befindet. Überprüfen Sie den Hintergrund und gehen Sie in der Beschreibung des Problems auf die darin aufgezeichneten Methoden zu. Überprüfen Sie die Kommentare, um die ausgeführten Aktionen zu verstehen. Wählen Sie unter Pull-Anforderungen den Link zur Pull-Anforderung neben der Bezeichnung Öffnen aus, um den Code zu überprüfen.
3. Überprüfen Sie in der Pull-Anforderung die Codeänderungen. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#). Übernehmen Sie Kommentare zur Pull-Anforderung, wenn Amazon Q einen der vorgeschlagenen Codes ändern soll. Seien Sie spezifisch, wenn Sie Kommentare für Amazon Q hinterlassen, um optimale Ergebnisse zu erzielen.

Wenn Sie beispielsweise die für erstellte Pull-Anforderung überprüfen **Create another mysfit named Quokkapus**, stellen Sie möglicherweise fest, dass die Beschreibung einen Tippfehler enthält. Sie könnten einen Kommentar für Amazon Q hinterlassen, der besagt: „Ändern Sie die Beschreibung, um den Tippfehler „needsa“ zu beheben, indem Sie ein Leerzeichen zwischen „needs“ und „a“ hinzufügen.“ Alternativ können Sie einen Kommentar hinterlassen, der Amazon Q anweist, die Beschreibung zu aktualisieren, und die gesamte überarbeitete Beschreibung angeben, damit sie integriert werden kann.

Wenn Sie Bilder für den neuen mysfit auf die Website hochgeladen haben, können Sie einen Kommentar für Amazon Q hinterlassen, um den mysfit mit Zeigern auf das Bild und die Miniaturansicht zu aktualisieren, die für den neuen mysfit verwendet werden sollen.

#### Note

Amazon Q antwortet nicht auf einzelne Kommentare. Amazon Q integriert nur Feedback, das in Kommentaren in Pull-Anfragen hinterlassen wurde, wenn Sie die Standardoption gewählt haben, nach jedem Schritt zur Genehmigung anzuhalten, als Sie das Problem erstellt haben.

4. (Optional) Nachdem Sie und andere Projektbenutzer alle Kommentare beibehalten haben, die Sie für Änderungen am Code wünschen, wählen Sie Revision erstellen aus, damit Amazon Q eine Revision der Pull-Anforderung erstellt, die die von Ihnen in den Kommentaren angeforderten Änderungen enthält. Der Fortschritt der Revisionserstellung wird von Amazon Q in Übersicht und nicht in Änderungen gemeldet. Stellen Sie sicher, dass Sie Ihren Browser aktualisieren, um die neuesten Updates von Amazon Q beim Erstellen der Revision anzuzeigen.

**Note**

Nur der Benutzer, der das Problem erstellt hat, kann eine Revision der Pull-Anforderung erstellen. Sie können nur eine Revision einer Pull-Anforderung anfordern. Stellen Sie sicher, dass Sie alle Probleme mit Kommentaren behoben haben und mit dem Inhalt der Kommentare zufrieden sind, bevor Sie Revision erstellen auswählen.

5. Für jede Pull-Anforderung in diesem Beispielprojekt wird ein Workflow ausgeführt. Stellen Sie sicher, dass Sie eine erfolgreiche Workflow-Ausführung sehen, bevor Sie die Pull-Anforderung zusammenführen. Sie können auch zusätzliche Workflows und Umgebungen erstellen, um den Code zu testen, bevor Sie ihn zusammenführen. Weitere Informationen finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#).
6. Wenn Sie mit der neuesten Revision der Pull-Anforderung zufrieden sind, wählen Sie Zusammenführen aus.

## Bereinigen von -Ressourcen

Wenn Sie dieses Tutorial abgeschlossen haben, sollten Sie die folgenden Aktionen ausführen, um alle Ressourcen zu bereinigen, die Sie während dieses Tutorials erstellt haben und die Sie nicht mehr benötigen.

- Heben Sie die Zuweisung von Amazon Q zu Problemen auf, die nicht mehr bearbeitet werden. Wenn Amazon Q seine Arbeit an einem Problem abgeschlossen hat oder keine Lösung finden konnte, stellen Sie sicher, dass Sie die Zuweisung von Amazon Q aufheben, um zu vermeiden, dass das maximale Kontingent für generative KI-Funktionen erreicht wird. Weitere Informationen finden Sie unter [Verwalten von Funktionen für generative KI](#) und [Preise für](#) .
- Verschieben Sie alle Probleme, bei denen die Arbeit abgeschlossen ist, nach Fertig .
- Wenn das Projekt nicht mehr benötigt wird, löschen Sie das Projekt.

## Tutorial: Erstellen einer Full-Stack-Anwendung mit zusammensetzbaren PDK-Blueprints

Amazon CodeCatalyst bietet eine Reihe verschiedener Blueprints, mit denen Sie schnell mit Ihren Projekten beginnen können. Ein mit einem Blueprint erstelltes Projekt umfasst die Ressourcen, die Sie benötigen, darunter ein Quell-Repository, Beispielquellcode, CI/CD-Workflows, Build-

und Testberichte sowie integrierte Tools zur Problemverfolgung. Manchmal möchten Sie jedoch möglicherweise ein Projekt schrittweise ausbauen oder einem vorhandenen Projekt, das anhand eines Blueprints erstellt wurde, Funktionen hinzufügen. Sie können dies auch mit Blueprints tun. Dieses Tutorial zeigt, wie Sie mit einem einzigen Blueprint beginnen können, der eine Grundlage bildet und es Ihnen ermöglicht, Ihren gesamten Projektcode in einem einzigen Repository zu speichern. Von dort aus haben Sie die Flexibilität, zusätzliche Ressourcen und Infrastruktur zu integrieren, indem Sie nach Belieben weitere Blueprints auf den ursprünglichen Blueprint anwenden. Mit dieser Bausteinmethode können Sie spezifische Anforderungen in mehreren Projekten erfüllen.

Dieses Tutorial zeigt, wie Sie mehrere AWS Project Development Kit (AWS PDK) -Blueprints zusammenstellen, um eine Anwendung zu erstellen, die aus einer React-Website, einer Smithy-API und der unterstützenden CDK-Infrastruktur für die Bereitstellung in AWS besteht. Das AWS PDK bietet Bausteine für gängige Muster sowie Entwicklungstools zur Verwaltung und Erstellung Ihrer Projekte. Weitere Informationen finden Sie im [AWS GitHub PDK-Quell-Repository](#).

Die folgenden PDK-Blueprints sind so konzipiert, dass sie miteinander verwendet werden können, um eine Anwendung auf zusammensetzbare Weise zu erstellen:

- [Monorepo](#) — Erstellt ein Projekt auf Stammebene, das die Interdependenzen zwischen Projekten innerhalb des Monorepo verwaltet. Das Projekt bietet auch Build-Caching und Abhängigkeitsvisualisierung.
- [Type Safe API](#) — Erstellt eine API, die entweder in [Smithy](#) oder [OpenAPI v3](#) definiert werden kann, und verwaltet die Generierung von Build-Time-Code, damit Sie Ihre API typsicher implementieren und mit ihr interagieren können. Stellt ein CDK-Konstrukt bereit, das die Bereitstellung Ihrer API für API Gateway verwaltet und die automatische Eingabvalidierung konfiguriert.
- [Cloudscape React-Website](#) — Erstellt eine React-basierte Website, die mit [Cloudscape](#) erstellt wurde und in Cognito Auth und (optional) Ihrer erstellten API vorintegriert ist, sodass Sie Ihre API sicher aufrufen können.
- [Infrastruktur](#) — Erstellt ein Projekt, das die gesamte CDK-bezogene Infrastruktur einrichtet, die für die Bereitstellung Ihrer Anwendung erforderlich ist. Es ist außerdem so vorkonfiguriert, dass es bei jedem Build ein Diagramm generiert, das auf Ihrem CDK-Code basiert.
- [DevOps](#) - Erstellt DevOps Workflows, die mit Konstrukten aus dem AWS Project Development Kit (AWS PDK) kompatibel sind.

Das Tutorial enthält auch Schritte zum Anzeigen der bereitgestellten Anwendung, zum Einladen anderer Benutzer zur Bearbeitung und zum Vornehmen von Änderungen am Code mit Pull-Requests,

die automatisch erstellt und für Ressourcen im verbundenen AWS-Konto bereitgestellt werden, wenn die Pull-Anfrage zusammengeführt wird.

Wenn Sie ein Projekt erstellen, das aus PDK-Blueprints besteht, wird Ihr Projekt mit den folgenden Ressourcen in einem CodeCatalyst Projekt erstellt:

- Ein als Monorepo konfiguriertes Quell-Repository.
- Ein Workflow, der statische Codeanalysen und Lizenzprüfungen durchführt sowie den Beispielcode erstellt und bereitstellt, wenn eine Änderung am Standardzweig vorgenommen wird. Jedes Mal, wenn Sie Änderungen am Code vornehmen, wird ein Architekturdiagramm generiert.
- Ein Issue-Board und ein Backlog, die Sie zur Planung und Nachverfolgung der Arbeit verwenden können.
- Eine Testberichtssuite mit automatisierten Berichten.

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie ein Monorepo-Projekt](#)
- [Schritt 2: Fügen Sie dem Projekt die Type Safe-API hinzu](#)
- [Schritt 3: Fügen Sie eine Cloudscape React-Website für das Projekt hinzu](#)
- [Schritt 4: Generieren Sie die Infrastruktur für die Bereitstellung der Anwendung in der AWS-Cloud](#)
- [Schritt 5: Richten Sie einen DevOps Workflow für die Bereitstellung Ihres Projekts ein](#)
- [Schritt 6: Bestätigen Sie den Release-Workflow und sehen Sie sich Ihre Website an](#)
- [Arbeiten Sie zusammen und iterieren Sie am PDK-Projekt](#)

## Voraussetzungen

Um ein Projekt zu erstellen und zu aktualisieren, müssen Sie die folgenden Aufgaben abgeschlossen haben: [Einrichten CodeCatalyst](#)

- Sie müssen über eine AWS Builder-ID verfügen, bei der Sie sich anmelden können CodeCatalyst.
- Gehören Sie einem Bereich an und lassen Sie sich in diesem Bereich die Rolle Space-Administrator oder Poweruser zuweisen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#), [Verwaltung von Space-Benutzern](#) und [Rolle „Space-Administrator“](#).

- Haben Sie ein AWS-Konto mit Ihrem Bereich verknüpft und verfügen Sie über die IAM-Rolle, die Sie bei der Registrierung erstellt haben. Bei der Registrierung haben Sie beispielsweise die Möglichkeit, eine Servicerolle mit einer Rollenrichtlinie namens CodeCatalystWorkflowDevelopmentRole- *SpaceName* -Rollenrichtlinie zu erstellen. Der Rolle wird ein Name CodeCatalystWorkflowDevelopmentRole-*spaceName* mit einer angehängten eindeutigen Kennung zugewiesen. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#). Die Schritte zum Erstellen der Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#).

## Schritt 1: Erstellen Sie ein Monorepo-Projekt

Beginnen Sie mit dem PDK-Monorepo-Blueprint, um Ihre Monorepo-Codebasis zu erstellen, die als Grundlage dient und es Ihnen ermöglicht, zusätzliche PDK-Blueprints hinzuzufügen.

Um ein Projekt mit dem PDK-Monorepo-Blueprint zu erstellen

1. [Öffnen Sie die Konsole unter https://codecatalyst.aws/ CodeCatalyst](https://codecatalyst.aws/) .
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
3. Wählen Sie im Space-Dashboard die Option Projekt erstellen aus.
4. Wählen Sie Mit einem Blueprint beginnen aus.
5. Wählen Sie den Blueprint PDK — Monorepo und wählen Sie dann Weiter aus.
6. Geben Sie unter Name your project den Namen ein, den Sie Ihrem Projekt zuweisen möchten, sowie die zugehörigen Ressourcennamen. Der Name muss in Ihrem Bereich eindeutig sein.
7. Gehen Sie unter Projektressourcen wie folgt vor:
  - a. Wählen Sie unter Primäre Programmiersprache eine Sprache aus, in der Sie Ihren Projektcode entwickeln möchten. Sie können zwischen TypeScript Java oder Python wählen.
  - b. Wählen Sie Code-Konfiguration
  - c. Geben Sie im Texteingabefeld Quell-Repository den Namen eines Quell-Repositorys ein, das ein neues Repository erstellen soll, oder wählen Sie aus einem vorhandenen verknüpften Repository aus. Das bestehende Repository muss leer sein. Weitere Informationen finden Sie unter [Ein Quell-Repository verlinken](#).

- d. (Optional) Wählen Sie im Dropdownmenü Package Manager einen Paketmanager aus. Dies ist nur erforderlich, wenn Sie es TypeScript als primäre Programmiersprache ausgewählt haben.
8. (Optional) Um eine Vorschau des Codes anzuzeigen, der auf der Grundlage der von Ihnen ausgewählten Projektparameter generiert wird, wählen Sie unter Projektvorschau generieren die Option Code anzeigen aus.
9. (Optional) Wählen Sie auf der Karte des Blueprints die Option Details anzeigen aus, um spezifische Details zum Blueprint anzuzeigen, z. B. einen Überblick über die Architektur des Blueprints, die erforderlichen Verbindungen und Berechtigungen sowie die Art der Ressourcen, die der Blueprint erstellt.
10. Wählen Sie Projekt erstellen aus, um Ihr Monorepo-Projekt zu erstellen. Das erstellte Projekt auf Stammebene verwaltet die Interdependenzen zwischen Projekten innerhalb des Monorepos und bietet Build-Caching und Abhängigkeitsmanagement.

Weitere Informationen zu Projekt-Blueprints finden Sie unter [Referenz zum Projekt-Blueprint](#)

Der Blueprint PDK — Monorepo generiert nur die Grundlage des Projekts. Um mithilfe des Blueprints eine funktionsfähige Anwendung zu erstellen, müssen Sie weitere PDK-Blueprints wie Type Safe API, Cloudscape React Website, Infrastructure oder hinzufügen. DevOps Im nächsten Schritt wenden Sie eine Type Safe-API auf das Projekt an.

## Schritt 2: Fügen Sie dem Projekt die Type Safe-API hinzu

Der PDK — Type Safe API-Blueprint ermöglicht es Ihnen, eine API mit Smithy oder OpenAPI v3 zu definieren. Es generiert Runtime-Pakete aus Ihrer API-Definition, die Clients für die Interaktion mit Ihrer API und serverseitigen Code für die Implementierung Ihrer API enthalten. Der Blueprint generiert außerdem ein CDK-Konstrukt mit Typsicherheit für jede API-Operation. Sie können den Blueprint auf ein vorhandenes PDK-Monorepo-Projekt anwenden, um dem Projekt API-Funktionen hinzuzufügen.

Um den API-Blueprint PDK — Type Safe anzuwenden

1. Wählen Sie im Navigationsbereich Ihres Monorepo-Projekts Blueprints und dann Blueprint anwenden aus.
2. Wählen Sie den Blueprint PDK — Type Safe API und dann Weiter aus.
3. Konfigurieren Sie unter Blueprint konfigurieren die Blueprint-Parameter:



- Wählen Sie unter Modellsprache die Sprache aus, in der das API-Modell definiert ist.
  - Geben Sie im Texteingabefeld Namespace einen Namespace für Ihre API ein.
  - Geben Sie im Texteingabefeld API-Name einen Namen für Ihre API ein.
  - Wählen Sie unter CDK-Sprache Ihre bevorzugte Sprache aus, in der die CDK-Infrastruktur geschrieben werden soll, in der die API bereitgestellt werden soll.
  - Wählen Sie das Dropdownmenü Handler-Sprache (n) und wählen Sie dann die Sprachen aus, in denen Sie Handler für API-Operationen implementieren möchten.
  - Wählen Sie das Dropdownmenü Dokumentationsformat (e) und wählen Sie dann die Formate aus, die Sie für die Generierung der API-Dokumentation verwenden möchten.
4. Überprüfen Sie auf der Registerkarte Codeänderungen die vorgeschlagenen Änderungen. Der Unterschied, der in einem Pull Request angezeigt wird, zeigt die Änderungen an deinem Projekt zum Zeitpunkt der Erstellung des Pull Requests.
  5. Wenn Sie mit den vorgeschlagenen Änderungen, die bei der Anwendung des Blueprints vorgenommen werden, zufrieden sind, wählen Sie Blueprint anwenden aus.

Nachdem eine Pull-Anfrage erstellt wurde, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @ Zeichen, gefolgt vom Namen der Datei, verwendest.

#### Note

Der Blueprint wird erst angewendet, wenn der Pull-Request genehmigt und zusammengeführt wurde. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

6. Wählen Sie in der Spalte Status die Option Ausstehende Pull-Anfrage für die Blueprint-Zeile PDK — Type Safe API aus und wählen Sie dann den Link der geöffneten Pull-Anfrage aus.
7. Wählen Sie Zusammenführen, wählen Sie Ihre bevorzugte Zusammenführungsstrategie und wählen Sie dann Zusammenführen aus, um die Änderungen aus dem angewendeten Blueprint zu übernehmen.

Nachdem der Pull-Request zusammengeführt wurde, wird in Ihrem Monorepo-Projekt ein neuer `packages/apis/myjdkapi` Ordner generiert, der den gesamten API-bezogenen Quellcode für Ihre konfigurierte Type Safe-API enthält.



8. Wählen Sie im Navigationsbereich Blueprints aus, um zu bestätigen, dass der Status der PDK — Type Safe API aktuell angezeigt wird.

## Schritt 3: Fügen Sie eine Cloudscape React-Website für das Projekt hinzu

Der Blueprint PDK — Cloudscape React Website generiert eine Website. Sie können einen optionalen Parameter (Type Safe APIs) zuordnen, um Ihre Website automatisch so zu konfigurieren, dass authentifizierte typsichere Clients sowie ein interaktiver API-Explorer zum Testen Ihrer verschiedenen APIs eingerichtet werden.

Um den Blueprint PDK — Cloudscape React Website anzuwenden

1. Wählen Sie im Navigationsbereich Ihres Monorepo-Projekts Blueprints und dann Blueprint anwenden aus.
2. Wählen Sie den Blueprint PDK — Cloudscape React Website und dann Weiter aus.
3. Konfigurieren Sie unter Blueprint konfigurieren die Blueprint-Parameter:
  - Geben Sie im Texteingabefeld für den Namen der Website einen Namen für Ihre Website ein.
  - Wählen Sie das Dropdownmenü Type Safe APIs und wählen Sie dann die API-Blueprints aus, die Sie in die Website integrieren möchten. Durch das Übergeben einer API werden authentifizierte Clients eingerichtet und die erforderlichen Abhängigkeiten, der API-Explorer und andere Funktionen hinzugefügt.
4. Überprüfen Sie auf der Registerkarte Codeänderungen die vorgeschlagenen Änderungen. Der Unterschied, der in einem Pull Request angezeigt wird, zeigt die Änderungen an deinem Projekt zum Zeitpunkt der Erstellung des Pull Requests.
5. Wenn Sie mit den vorgeschlagenen Änderungen, die bei der Anwendung des Blueprints vorgenommen werden, zufrieden sind, wählen Sie Blueprint anwenden aus.

Nachdem eine Pull-Anfrage erstellt wurde, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @ Zeichen, gefolgt vom Namen der Datei, verwendest.

**Note**

Der Blueprint wird erst angewendet, wenn der Pull-Request genehmigt und zusammengeführt wurde. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

- Wählen Sie in der Spalte Status die Option Ausstehende Pull-Anfrage für die Blueprint-Zeile PDK — Cloudscape React Website aus und wählen Sie dann den Link der geöffneten Pull-Anfrage aus.
- Wählen Sie Zusammenführen, wählen Sie Ihre bevorzugte Merge-Strategie und wählen Sie dann Merge, um die Änderungen aus dem angewendeten Blueprint zu übernehmen.

Nachdem der Pull-Request zusammengeführt wurde, wird in Ihrem Monorepo-Projekt ein neuer `packages/websites/my-website-name` Ordner generiert, der den gesamten Quellcode für Ihre neue Website enthält.

- Wählen Sie im Navigationsbereich Blueprints aus, um zu bestätigen, dass der Status der PDK — Cloudscape React-Website aktuell angezeigt wird.

Als Nächstes wenden Sie den PDK — Infrastructure Blueprint an, um die Infrastruktur für die Bereitstellung Ihrer Website in der AWS-Cloud zu generieren.


## Schritt 4: Generieren Sie die Infrastruktur für die Bereitstellung der Anwendung in der AWS-Cloud

Der PDK — Infrastructure Blueprint richtet ein Paket ein, das Ihren gesamten CDK-Code für die Bereitstellung Ihrer Website und API enthält. Es bietet standardmäßig auch die Generierung von Diagrammen und die Konformität mit dem Prototyping-Nag Pack.

Um den PDK — Infrastructure Blueprint anzuwenden

- Wählen Sie im Navigationsbereich Ihres Monorepo-Projekts Blueprints und dann Blueprint anwenden aus.
- Wählen Sie den Blueprint PDK — Infrastruktur und dann Weiter aus.
- Konfigurieren Sie unter Blueprint konfigurieren die Blueprint-Parameter:
  - Wählen Sie unter CDK-Sprache die Sprache aus, mit der Sie Ihre Infrastruktur entwickeln möchten.


- Geben Sie im Texteingabefeld Stackname einen Namen des für Ihren Blueprint generierten CloudFormation Stacks ein.

 Note

Notieren Sie sich diesen Stack-Namen für den nächsten Schritt, in dem Sie einen DevOps Workflow einrichten werden.

- Wählen Sie das Dropdownmenü Type Safe APIs und wählen Sie dann die API-Blueprints aus, die Sie in die Website integrieren möchten.
  - Wählen Sie das Dropdownmenü Cloudscape React TS Websites und wählen Sie dann Website-Blueprints aus, die Sie in Ihrer Infrastruktur bereitstellen möchten (z. B. PDK — Cloudscape React Website).
4. Überprüfen Sie auf der Registerkarte Codeänderungen die vorgeschlagenen Änderungen. Der Unterschied, der in einem Pull Request angezeigt wird, zeigt die Änderungen an deinem Projekt zum Zeitpunkt der Erstellung des Pull Requests.
  5. Wenn Sie mit den vorgeschlagenen Änderungen, die bei der Anwendung des Blueprints vorgenommen werden, zufrieden sind, wählen Sie **Blueprint anwenden** aus.

Nachdem eine Pull-Anfrage erstellt wurde, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @ Zeichen, gefolgt vom Namen der Datei, verwendest.

 Note

Der Blueprint wird erst angewendet, wenn der Pull-Request genehmigt und zusammengeführt wurde. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

6. Wählen Sie in der Spalte Status die Option **Ausstehende Pull-Anfrage** für die Zeile **PDK — Infrastruktur-Blueprint** aus und wählen Sie dann den Link der geöffneten Pull-Anfrage aus.
7. Wählen Sie **Zusammenführen**, wählen Sie Ihre bevorzugte Zusammenführungsstrategie und wählen Sie dann **Zusammenführen** aus, um die Änderungen aus dem angewendeten Blueprint zu übernehmen.

Nachdem die Pull-Anfrage zusammengeführt wurde, wird in Ihrem Monorepo-Projekt ein neuer `packages/infra` Ordner generiert, der die Infrastruktur enthält, mit der Ihr Projekt in der AWS-Cloud bereitgestellt wird.

8. Wählen Sie im Navigationsbereich Blueprints aus, um zu bestätigen, dass der Status der PDK-Infrastruktur aktuell angezeigt wird.

Als Nächstes wenden Sie den DevOpsPDK-Blueprint an, um Ihre Anwendung bereitzustellen.

## Schritt 5: Richten Sie einen DevOps Workflow für die Bereitstellung Ihres Projekts ein

Der DevOpsPDK-Blueprint generiert die erforderlichen DevOps Workflows, um Ihr Projekt mithilfe des AWS-Kontos und der in der Konfiguration angegebenen AWS-Rolle zu erstellen und bereitzustellen.

Um den PDK-Blueprint anzuwenden DevOps

1. Wählen Sie im Navigationsbereich Ihres Monorepo-Projekts Blueprints und dann Blueprint anwenden aus.
2. Wählen Sie den DevOpsPDK-Blueprint und dann Weiter aus.
3. Konfigurieren Sie unter Blueprint konfigurieren die Blueprint-Parameter:
  - Wählen Sie Bootstrap CDK in der aktuellen Umgebung aus.
  - Geben Sie im Texteingabefeld Stackname den Namen des CloudFormation Stacks ein, den Sie bereitstellen möchten. Dieser sollte mit dem Stacknamen übereinstimmen, der [Schritt 4: Generieren Sie die Infrastruktur für die Bereitstellung der Anwendung in der AWS-Cloud](#) für den PDK-Infrastruktur-Blueprint konfiguriert ist.
  - Wählen Sie das Dropdownmenü AWS-Kontoverbindung und dann das AWS-Konto aus, das Sie für die Ressourcen verwenden möchten. Weitere Informationen finden Sie unter [Ein AWS-Konto zu einem Leerzeichen hinzufügen](#).
  - Wählen Sie im Dropdownmenü Die Rolle aus, die für die Bereitstellung Ihrer Anwendung verwendet werden soll, und wählen Sie dann die IAM-Rolle aus, die Sie für die Bereitstellung Ihrer Projektanwendung verwenden möchten.

**Note**

Wenn Sie eine IAM-Rolle erstellen, beschränken Sie die Rolle auf die aktuelle Rolle, die `SourceArn` Sie in den `ProjectID` Projekteinstellungen finden. Weitere Informationen finden Sie unter [Grundlegendes zur `CodeCatalystWorkflowDevelopmentRole-spaceName` Servicerolle](#).

- Wählen Sie das Dropdownmenü `Region` und dann die Region aus, in der Sie Ihr Monorepo-Projekt bereitstellen möchten. Die Bereitstellung funktioniert nur in Regionen, in denen die erforderlichen AWS-Services verfügbar sind. Weitere Informationen finden Sie unter [AWS-Services nach Regionen](#).
4. Überprüfen Sie auf der Registerkarte Codeänderungen die vorgeschlagenen Änderungen. Der Unterschied, der in einem Pull Request angezeigt wird, zeigt die Änderungen an deinem Projekt zum Zeitpunkt der Erstellung des Pull Requests.
  5. Wenn Sie mit den vorgeschlagenen Änderungen, die bei der Anwendung des Blueprints vorgenommen werden, zufrieden sind, wählen Sie `Blueprint anwenden` aus.

Nachdem eine Pull-Anfrage erstellt wurde, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das `@` Zeichen, gefolgt vom Namen der Datei, verwendest.


**Note**

Der Blueprint wird erst angewendet, wenn der Pull-Request genehmigt und zusammengeführt wurde. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

6. Wählen Sie in der Spalte `Status` die Option `Ausstehende Pull-Anfrage` für die Zeile `PDK — Infrastruktur-Blueprint` aus und wählen Sie dann den Link der geöffneten Pull-Anfrage aus.
7. Wählen Sie `Zusammenführen`, wählen Sie Ihre bevorzugte Zusammenführungsstrategie und wählen Sie dann `Zusammenführen` aus, um die Änderungen aus dem angewendeten Blueprint zu übernehmen.

Nachdem der Pull-Request zusammengeführt wurde, wird ein neuer `.codecatalyst/workflows` Ordner in Ihrem Monorepo-Projekt generiert.

- Wählen Sie im Navigationsbereich Blueprints aus, um zu bestätigen, dass der Status des PDK — Aktuell DevOps angezeigt wird.

 Note

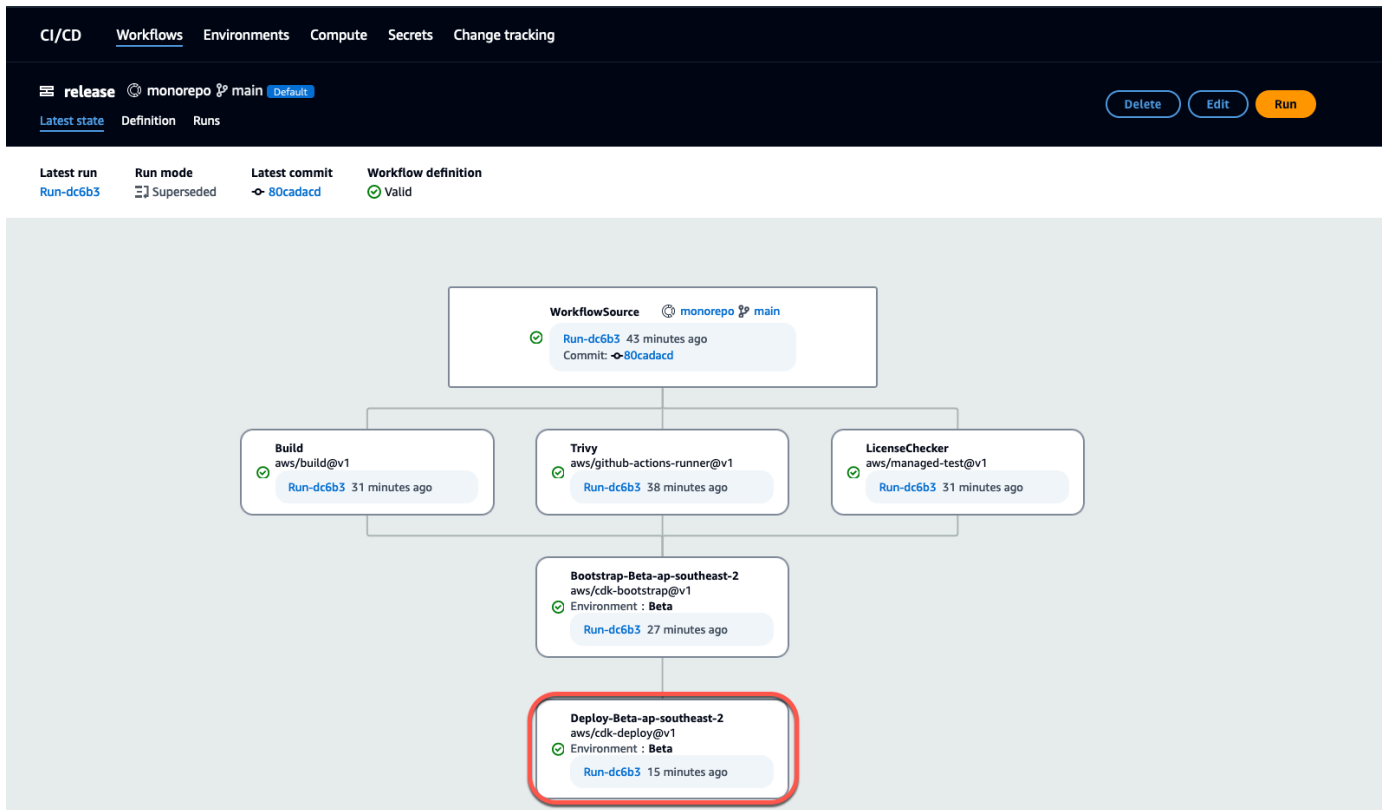
Der DevOpsPDK-Blueprint und alle nachfolgenden Änderungen an PDK-Blueprints werden ab diesem Zeitpunkt deutlich langsamer sein, da hinter den Kulissen Sperrdateien generiert werden, um sicherzustellen, dass Builds und Deployments in future wiederholbar sind. Es generiert Sperrdateien für alle Pakete in allen unterstützten Sprachen.

## Schritt 6: Bestätigen Sie den Release-Workflow und sehen Sie sich Ihre Website an

Sobald Sie die vorherigen Schritte abgeschlossen haben, können Sie den Release-Workflow bestätigen, um sicherzustellen, dass das Projekt erstellt wird.

Um den Release-Workflow zu bestätigen und Ihre Website aufzurufen

- Wählen Sie im Navigationsbereich Ihres Monorepo-Projekts CI/CD und dann Workflows aus.
- Wählen Sie für den Release-Workflow den zuletzt ausgeführten Workflow aus, um die Details anzuzeigen. Weitere Informationen finden Sie unter [Anzeigen des Status und der Details einer einzelnen Ausführung](#).
- Nachdem die Workflow-Ausführung erfolgreich abgeschlossen wurde, wählen Sie die letzte Aktion im Workflow aus (z. B. Deploy-B eta-ap-souteast -2), und wählen Sie dann Variablen aus.



4. Zeigen Sie die bereitgestellte Website an, indem Sie den Link in der Variablentabelle (z. B. ***myPDKApi websiteDistributionDomain NameXXXX***) kopieren und in ein neues **Browserfenster** einfügen.

## Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds


Logs

Summary

Configuration

**Variables**

### Output variables (7)

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://iczdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityUserPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	 arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7



Sie benötigen ein Amazon Cognito Konto, um sich auf Ihrer Website anzumelden. Standardmäßig ist der Benutzerpool nicht so eingerichtet, dass eine Selbstregistrierung möglich ist.

- a. Navigieren Sie zur [AWS Cognito-Konsole](#).
- b. *Wählen Sie in der Tabelle **Benutzerpools** den Namen des **Benutzerpools** aus, der dem Benutzerpool entspricht, der mit dem **DevOpsPDK-Blueprint** erstellt wurde. Dieser befindet sich in der **Variablentabelle** (z. B. **infra calculate XXXXX**). **betaUserIdentityinfra** betaUserIdentity IdentityPoolId Weitere Informationen finden Sie unter [Erste Schritte mit Benutzerpools](#).*

## Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds


Logs

Summary

Configuration

Variables

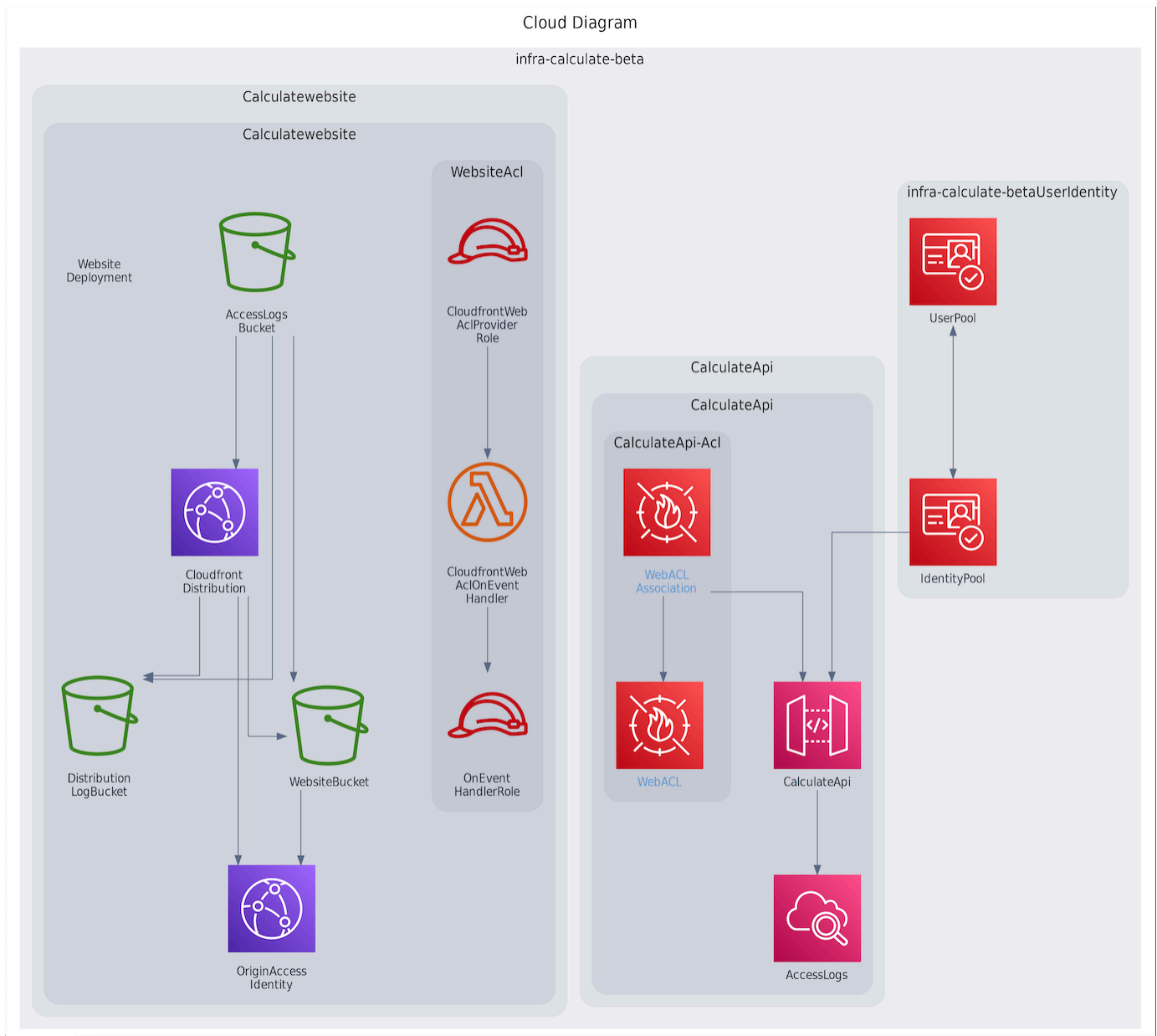
## Output variables (7)

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://icfdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	 arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

- c. Wählen Sie Create user (Benutzer erstellen) aus.
  - d. Konfigurieren Sie die Parameter für Benutzerinformationen:
    - Wählen Sie unter Einladungsnachricht die Option E-Mail-Einladung senden aus.
    - Geben Sie im Texteingabefeld Benutzername einen Benutzernamen ein.
    - Geben Sie im Texteingabefeld E-Mail-Adresse einen Benutzernamen ein.
    - Wählen Sie unter Temporäres Passwort die Option Passwort generieren aus.
  - e. Wählen Sie Create user (Benutzer erstellen) aus.
  - f. Navigieren Sie zu dem E-Mail-Konto, das Sie für die Parameter Benutzerinformationen eingegeben haben, und öffnen Sie eine E-Mail mit einem temporären Passwort. Notieren Sie sich das Passwort.
  - g. Gehen Sie zurück zur bereitgestellten Website, geben Sie den von Ihnen erstellten Benutzernamen und das temporäre Passwort ein, das Sie erhalten haben, und wählen Sie dann Anmelden aus.
5. (Optional) Nachdem die Workflow-Ausführung erfolgreich abgeschlossen wurde, können Sie auch das generierte Diagramm anzeigen. Wählen Sie die Registerkarte Artefakte unter CodeCatalyst, wählen Sie Herunterladen für die Diagrammzeile aus, und öffnen Sie dann die heruntergeladene (n) Datei (en).

The screenshot shows the 'Artifacts' tab for a workflow named 'Run-ef953'. The workflow status is 'Succeeded'. The artifacts table is as follows:

Artifact name	Files	Produced by	Consumed by
Built	<a href="#">Download</a>	Build	Bootstrap-Beta-ap-southeast-2 Deploy-Beta-ap-southeast-2
<b>Diagram</b>	<a href="#">Download</a>	Build	-
bdd254b65baac169f6ac50e8175ce6d930c1fcb086dec59808d3e0170ae2291d_report	<a href="#">Download</a>	Build	-
a093422585a8a4cb763d89a0fa8e76744a80830fe24724c7e7943a50ec479240_report	<a href="#">Download</a>	Trivy	-



## Arbeiten Sie zusammen und iterieren Sie am PDK-Projekt

Nachdem Ihr Projekt eingerichtet ist, können Sie Änderungen am Quellcode vornehmen. Sie können auch andere Space-Mitglieder einladen, an dem Projekt zu arbeiten. PDK-Blueprints ermöglichen es Ihnen, Ihre Anwendung iterativ zu erstellen und nur das hinzuzufügen, was Sie benötigen, wenn Sie es benötigen, und behalten gleichzeitig die volle Kontrolle über die Konfiguration der einzelnen Blueprints.

### Themen

- [Schritt 1: Laden Sie Mitglieder zu Ihrem Projekt ein](#)
- [Schritt 2: Erstellen Sie Probleme, um zusammenzuarbeiten und die Arbeit nachzuverfolgen](#)
- [Schritt 3: Sehen Sie sich Ihr Quell-Repository an](#)
- [Schritt 4: Erstellen Sie eine Entwicklungsumgebung und nehmen Sie Codeänderungen vor](#)
- [Schritt 5: Codeänderungen per Push übertragen und zusammenführen](#)

## Schritt 1: Laden Sie Mitglieder zu Ihrem Projekt ein

Sie können die Konsole verwenden, um Benutzer zu Ihrem Projekt einzuladen. Sie können Mitglieder Ihres Bereichs einladen oder Namen von außerhalb Ihres Bereichs hinzufügen.

Um Benutzer zu Ihrem Projekt einzuladen, müssen Sie mit der Rolle Projektadministrator oder Space-Administrator angemeldet sein.

Sie müssen keinen Benutzer mit der Rolle eines Space-Administrators zu Ihrem Projekt einladen, da dieser bereits impliziten Zugriff auf alle Projekte im Space hat.

Wenn Sie einen Benutzer zu Ihrem Projekt einladen (ohne ihm die Rolle des Space-Administrators zuzuweisen), wird der Benutzer in der Tabelle Projektmitglieder unter Projekte und in der Tabelle Projektmitglieder unter Spaces angezeigt.

Um ein Mitglied über den Tab Projekteinstellungen zu Ihrem Projekt einzuladen

1. Navigiere zu deinem Projekt.

### Tip

In der oberen Navigationsleiste können Sie auswählen, welches Projekt angezeigt werden soll.

2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Wählen Sie die Registerkarte Mitglieder.
4. Wählen Sie unter Projektmitglieder die Option Neues Mitglied einladen aus.
5. Geben Sie die E-Mail-Adresse des neuen Mitglieds ein, wählen Sie die Rolle für dieses Mitglied aus und wählen Sie dann Einladen aus. Weitere Informationen zu Rollen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

Um ein Mitglied von der Projektübersichtsseite aus zu Ihrem Projekt einzuladen

1. Navigiere zu deinem Projekt.

 Tip

In der oberen Navigationsleiste können Sie auswählen, welches Projekt angezeigt werden soll.


2. Wählen Sie die Schaltfläche Mitglieder +.
3. Geben Sie die E-Mail-Adresse des neuen Mitglieds ein, wählen Sie die Rolle für dieses Mitglied aus und wählen Sie dann Einladen aus. Weitere Informationen zu Rollen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

## Schritt 2: Erstellen Sie Probleme, um zusammenzuarbeiten und die Arbeit nachzuverfolgen

CodeCatalyst hilft Ihnen dabei, Funktionen, Aufgaben, Fehler und alle anderen Arbeiten im Zusammenhang mit Problemen im Zusammenhang mit Ihrem Projekt nachzuverfolgen. Sie können Probleme erstellen, um die benötigte Arbeit und Ideen nachzuverfolgen. Wenn du ein Problem erstellst, wird es standardmäßig zu deinem Backlog hinzugefügt. Du kannst Probleme in ein Board verschieben, in dem du die laufenden Arbeiten nachverfolgen kannst. Du kannst ein Problem auch einem bestimmten Projektmitglied zuweisen. Erstellen Sie in diesem Schritt ein Problem, um Änderungen an Ihrem PDK-Projekt vorzunehmen.

Um ein Problem zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Monorepo-Projekt, in dem Sie ein Problem erstellen möchten.
3. Wählen Sie auf der Projekt-Startseite die Option Problem erstellen aus. Wählen Sie alternativ im Navigationsbereich Probleme aus.
4. Wählen Sie Problem erstellen aus.

 Note

Sie können Probleme auch direkt hinzufügen, wenn Sie eine Rasteransicht verwenden.

5. Geben Sie einen Titel für das Problem ein.

6. (Optional) Geben Sie eine Beschreibung ein. Geben Sie für dieses Problem die folgende Beschreibung ein: `a change in the src/mysfit_data.json file`. Sie können Markdown verwenden, um Formatierungen hinzuzufügen.
7. (Optional) Wählen Sie einen Status, eine Priorität und eine Schätzung für das Problem aus.
8. (Optional) Fügen Sie ein vorhandenes Label hinzu oder erstellen Sie ein neues Label und fügen Sie es hinzu, indem Sie `+ Label hinzufügen` wählen.
  - a. Um ein vorhandenes Label hinzuzufügen, wählen Sie das Label aus der Liste aus. Sie können einen Suchbegriff in das Feld eingeben, um alle Labels im Projekt zu durchsuchen, die diesen Begriff enthalten.
  - b. Um ein neues Label zu erstellen und es hinzuzufügen, geben Sie den Namen des Labels, das Sie erstellen möchten, in das Suchfeld ein und drücken Sie die Eingabetaste.
9. (Optional) Fügen Sie einen Beauftragten hinzu, indem Sie `+ Bevollmächtigten hinzufügen` wählen. Sie können sich schnell selbst als Beauftragten hinzufügen, indem Sie `+ Mich hinzufügen` wählen.

 Tip

Sie können Amazon Q ein Problem zuweisen, damit Amazon Q versucht, das Problem zu lösen. Weitere Informationen finden Sie unter [Tutorial: Verwenden CodeCatalyst generativer KI-Funktionen zur Beschleunigung Ihrer Entwicklungsarbeit](#).

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).

10. (Optional) Fügen Sie ein vorhandenes benutzerdefiniertes Feld hinzu oder erstellen Sie ein neues benutzerdefiniertes Feld. Probleme können mehrere benutzerdefinierte Felder haben.
  - a. Um ein vorhandenes benutzerdefiniertes Feld hinzuzufügen, wählen Sie das benutzerdefinierte Feld aus der Liste aus. Sie können einen Suchbegriff in das Feld eingeben, um alle benutzerdefinierten Felder im Projekt zu durchsuchen, die diesen Begriff enthalten.
  - b. Um ein neues benutzerdefiniertes Feld zu erstellen und hinzuzufügen, geben Sie den Namen des benutzerdefinierten Felds, das Sie erstellen möchten, in das Suchfeld ein und drücken Sie die Eingabetaste. Wählen Sie dann den Typ des benutzerdefinierten Felds aus, das Sie erstellen möchten, und legen Sie einen Wert fest.

11. Wählen Sie „Problem erstellen“. In der unteren rechten Ecke wird eine Benachrichtigung angezeigt: Wenn das Problem erfolgreich erstellt wurde, wird eine Bestätigungsmeldung angezeigt, dass das Problem erfolgreich erstellt wurde. Wenn das Problem nicht erfolgreich erstellt wurde, wird eine Fehlermeldung mit dem Grund für den Fehler angezeigt. Sie können dann „Erneut versuchen“ wählen, um das Problem zu bearbeiten und erneut zu erstellen, oder „Verwerfen“ wählen, um das Problem zu verwerfen. Bei beiden Optionen wird die Benachrichtigung verworfen.

 Note

Du kannst einen Pull-Request nicht mit einem Issue verknüpfen, wenn du ihn erstellst. Du kannst ihn jedoch [bearbeiten, nachdem du ihn](#) erstellt hast, um Links zu Pull Requests hinzuzufügen.

Weitere Informationen finden Sie unter [Probleme in CodeCatalyst](#).

### Schritt 3: Sehen Sie sich Ihr Quell-Repository an

Sie können die mit einem Projekt verknüpften Quell-Repositorys in Amazon CodeCatalyst einsehen. Für Quell-Repositorys in CodeCatalyst bietet die Übersichtsseite für ein Repository einen schnellen Überblick über Informationen und Aktivitäten in diesem Repository, einschließlich:

- Die Beschreibung des Repositorys, falls vorhanden
- Die Anzahl der Branches im Repository
- Die Anzahl der offenen Pull-Requests für das Repository
- Die Anzahl verwandter Workflows für das Repository
- Die Dateien und Ordner in der Standardverzweigung oder in der Verzweigung, die Sie wählen
- Der Titel, der Autor und das Datum der letzten Übertragung in den angezeigten Zweig
- Der Inhalt der README.md-Datei wurde in Markdown gerendert, sofern eine README.md-Datei enthalten ist

Diese Seite bietet auch Links zu den Commits, Branches und Pull-Requests für das Repository sowie eine schnelle Möglichkeit, einzelne Dateien zu öffnen, anzusehen und zu bearbeiten.



**Note**

Sie können diese Informationen zu verknüpften Repositorys nicht in der CodeCatalyst Konsole einsehen. Um Informationen über verknüpfte Repositorys anzuzeigen, wählen Sie den Link in der Liste der Repositorys, um das Repository in dem Dienst zu öffnen, der es hostet.

Um zu den Quell-Repositorys für ein Projekt zu navigieren

1. Navigieren Sie zu Ihrem Projekt und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie auf der Übersichtsseite für Ihr Projekt das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen.
  - Wählen Sie im Navigationsbereich Code und dann Quell-Repositorys aus. Wählen Sie unter Quell-Repositorys den Namen des Repositorys aus der Liste aus. Sie können die Liste der Repositorys filtern, indem Sie einen Teil des Repository-Namens in die Filterleiste eingeben.
2. Sehen Sie sich auf der Startseite des Repositorys den Inhalt des Repositorys und Informationen zu den zugehörigen Ressourcen an, z. B. die Anzahl der Pull-Requests und Workflows. Standardmäßig werden die Inhalte für den Standard-Branch angezeigt. Sie können die Ansicht ändern, indem Sie einen anderen Zweig aus der Dropdownliste auswählen.

**Tip**


Sie können auch schnell zu den Repositorys Ihres Projekts navigieren, indem Sie auf der Seite mit der Projektzusammenfassung die Option Projektcode anzeigen auswählen.

## Schritt 4: Erstellen Sie eine Entwicklungsumgebung und nehmen Sie Codeänderungen vor

Erstellen Sie in diesem Schritt eine Entwicklungsumgebung und nehmen Sie Codeänderungen vor, die dann mit dem Hauptzweig zusammengeführt werden. Dieses Tutorial führt Sie durch ein einfaches AWS-PDK-Projekt, Sie können aber auch einem komplexeren Beispiel folgen, das im [AWS GitHub PDK-Repository](#) bereitgestellt wird.

Um eine Entwicklungsumgebung mit einem neuen Zweig zu erstellen


1. Führen Sie im Navigationsbereich Ihres Monorepo-Projekts einen der folgenden Schritte aus:
  - Wählen Sie Überblick und navigieren Sie dann zum Abschnitt Meine Entwicklungsumgebungen.
  - Wählen Sie Code und dann Entwicklungsumgebungen aus.
  - Wählen Sie Code, wählen Sie Quell-Repositories und dann das Monorepo-Repository aus, für das Sie eine Entwicklungsumgebung erstellen möchten.
2. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
3. Wählen Sie Repository klonen.
4. Wählen Sie das Repository aus, das geklont werden soll, wählen Sie In neuem Zweig arbeiten, geben Sie einen Branch-Namen in das Feld Branch-Name ein und wählen Sie im Drop-down-Menü Branch erstellen von einem Branch aus, aus dem der neue Branch erstellt werden soll.

 Note

Wenn Sie eine Entwicklungsumgebung von der Seite Quell-Repositorys oder von einem bestimmten Quell-Repository aus erstellen, müssen Sie kein Repository auswählen. Die Entwicklungsumgebung wird aus dem Quell-Repository erstellt, das Sie auf der Seite Quell-Repositorys ausgewählt haben.

5. (Optional) Geben Sie unter Alias — optional einen Alias für die Entwicklungsumgebung ein.
6. (Optional) Wählen Sie die Schaltfläche „Konfiguration bearbeiten“, um die Rechen-, Speicher- oder Timeout-Konfiguration der Entwicklungsumgebung zu bearbeiten.
7. (Optional) Wählen Sie in Amazon Virtual Private Cloud (Amazon VPC) — optional im Drop-down-Menü eine VPC-Verbindung aus, die Sie mit Ihrer Entwicklungsumgebung verknüpfen möchten.

Wenn eine Standard-VPC für Ihren Bereich festgelegt ist, werden Ihre Entwicklungsumgebungen mit dieser VPC verbunden ausgeführt. Sie können dies überschreiben, indem Sie eine andere VPC-Verbindung zuordnen. Beachten Sie außerdem, dass VPC-verbundene Entwicklungsumgebungen das AWS Toolkit nicht unterstützen.

 Note

Wenn Sie eine Entwicklungsumgebung mit einer VPC-Verbindung erstellen, wird innerhalb der VPC eine neue Netzwerkschnittstelle erstellt. CodeCatalyst interagiert mit

dieser Schnittstelle über die zugehörige VPC-Rolle. Stellen Sie außerdem sicher, dass Ihr IPv4-CIDR-Block nicht für den **172.16.0.0/12** IP-Adressbereich konfiguriert ist.

- Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, wird in der Statusspalte Entwicklungsumgebung die Meldung Wird gestartet angezeigt, und in der Statusspalte wird nach Erstellung der Entwicklungsumgebung ausgeführt angezeigt.

Nachdem Ihre Entwicklungsumgebung ausgeführt wurde, können Sie mit Ihrer generierten Beispielanwendung arbeiten, CodeCatalyst indem Sie Änderungen am Code mit Pull-Requests vornehmen, die automatisch erstellt und für Ressourcen im verbundenen AWS-Konto bereitgestellt werden, wenn die Pull-Anfrage zusammengeführt wird. Das Monorepo sendet eine Dev-Datei, sodass alle erforderlichen globalen Abhängigkeiten und Laufzeiten automatisch vorhanden sind.

Um den Code in Ihrem Projekt zu ändern

- Navigieren Sie in einem funktionierenden Terminal Ihrer Entwicklungsumgebung zu Ihrem Monorepo-Projekt und installieren Sie dann Ihre Projektabhängigkeiten, indem Sie den folgenden Befehl ausführen:

```
npx projen install
```

- Navigieren Sie zu `packages/apis/myjdkapi/model/src/main/smithy/operations/say-hello.smithy`, das einen API-Beispielvorgang definiert. In diesem Tutorial erstellen Sie eine einfache Calculate Operation, bei der zwei Zahlen addiert werden. Nehmen Sie eine Änderung am Code vor, um diese Operation zu definieren, einschließlich ihrer Eingabe und Ausgabe.

Beispiel:

```
$version: "2"
namespace com.aws

@http(method: "POST", uri: "/calculate")
@handler(language: "typescript")
operation Calculate {
  input := {
    @required
    numberA: Integer
    @required
    numberB: Integer
```

```
    }
    output := {
        @required
        result: Integer
    }
}
```

Das `@handler` Merkmal teilt der Type Safe API mit, dass Sie diesen Vorgang als eingeschriebenen AWS-Lambda-Handler implementieren werden. TypeScript Die Type Safe-API generiert einen Stub für diesen Vorgang, in dem Sie ihn implementieren können. TypeScript Das `@required` Merkmal wird hinzugefügt, was bedeutet, dass es zur Laufzeit durch das bereitgestellte API-Gateway durchgesetzt wird. Weitere Informationen finden Sie in der [Smithy-Dokumentation](#).

3. Benennen Sie den `/say-hello.smithy` Dateinamen um, sodass er Ihren Codeänderungen entspricht (z. B. `calculate.smithy`).
4. Navigieren Sie zu dem `packages/apis/myjdkapi/model/src/main/smithy/main.smithy` und nehmen Sie eine Änderung am Code vor, um den Vorgang einzuleiten. Sie können die in der definierte `Calculate` Operation verfügbar machen, `/calculate.smithy` indem Sie sie im `operations` Feld dieser Datei auflisten.

Beispiel:

```
$version: "2"
namespace com.aws

use aws.protocols#restJson1

/// A sample smithy api
@restJson1
service MyPDKApi {
    version: "1.0"
    operations: [Calculate]
    errors: [
        BadRequestError
        NotAuthorizedError
        InternalFailureError
    ]
}
```

5. Erstellen Sie die Änderungen, indem Sie den folgenden Befehl ausführen:

```
npx projen build
```

**Note**

Optional können Sie ein `--parallel X` Flag angeben, das den Build auf die X Kerne verteilt.

Da das `@handler` Merkmal hinzugefügt wurde, werden nach Abschluss des Builds die folgenden Dateien generiert:

- `/packages/apis/myjdkapi/handlers/typescript/src/calculate.ts`
- `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts`

6. Navigieren Sie zum `packages/apis/myjdkapi/handlers/typescript/src/calculate.ts` und nehmen Sie Änderungen am Code vor. Diese Datei ist der Server-Handler, der für die API aufgerufen wird.

```
import {
  calculateHandler,
  CalculateChainedHandlerFunction,
  INTERCEPTORS,
  Response,
  LoggingInterceptor,
} from 'myjdkapi-typescript-runtime';

/**
 * Type-safe handler for the Calculate operation
 */
export const calculate: CalculateChainedHandlerFunction = async (request) => {
  LoggingInterceptor.getLogger(request).info('Start Calculate Operation');

  const { input } = request;

  return Response.success({
    result: input.body.numberA + input.body.numberB,
  });
};

/**
```

```
* Entry point for the AWS Lambda handler for the Calculate operation.  
* The calculateHandler method wraps the type-safe handler and manages marshalling  
inputs and outputs  
*/  
export const handler = calculateHandler(...INTERCEPTORS, calculate);
```

7. Navigieren Sie zu der `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts` Datei und nehmen Sie Änderungen am Code vor, um die Komponententests zu aktualisieren.

Beispiel:

```
import {  
  CalculateChainedRequestInput,  
  CalculateResponseContent,  
} from 'myjdkapi-typescript-runtime';  
import {  
  calculate,  
} from '../src/calculate';  
  
// Common request arguments  
const requestArguments = {  
  chain: undefined as never,  
  event: {} as any,  
  context: {} as any,  
  interceptorContext: {  
    logger: {  
      info: jest.fn(),  
    },  
  },  
}  
} satisfies Omit<CalculateChainedRequestInput, 'input'>;  
  
describe('Calculate', () => {  
  
  it('should return correct sum', async () => {  
    const response = await calculate({  
      ...requestArguments,  
      input: {  
        requestParameters: {},  
        body: {  
          numberA: 1,  
          numberB: 2  
        }  
      }  
    })  
  })  
})
```

```

    },
  });

  expect(response.statusCode).toBe(200);
  expect((response.body as CalculateResponseContent).result).toEqual(3);
});

});

```

8. Navigieren Sie zu der `/packages/infra/main/src/constructs/apis/myjdkapi.ts` Datei und nehmen Sie Änderungen am Code vor, um eine Integration für den Calculate Vorgang in Ihrer CDK-Infrastruktur hinzuzufügen. Das API-Konstrukt verfügt über eine Integrationseigenschaft, mit der Sie die zuvor hinzugefügte Implementierung übergeben können. Da Sie das `@handler` Merkmal in Ihrem Smithy-Modell für die Calculate Operation verwenden, können Sie das generierte `CalculateFunction` CDK-Konstrukt, das vorkonfiguriert ist, verwenden, um auf Ihre Handler-Implementierung zu verweisen.

Beispiel:

```

import { UserIdentity } from "@aws/pdk/identity";
import { Authorizers, Integrations } from "@aws/pdk/type-safe-api";
import { Stack } from "aws-cdk-lib";
import { Cors } from "aws-cdk-lib/aws-apigateway";
import {
  AccountPrincipal,
  AnyPrincipal,
  Effect,
  PolicyDocument,
  PolicyStatement,
} from "aws-cdk-lib/aws-iam";
import { Construct } from "constructs";
import { Api, CalculateFunction } from "calculateapi-typescript-infra";

/**
 * Api construct props.
 */
export interface CalculateApiProps {
  /**
   * Instance of the UserIdentity.
   */
  readonly userIdentity: UserIdentity;
}

```

```
/**
 * Infrastructure construct to deploy a Type Safe API.
 */
export class CalculateApi extends Construct {
  /**
   * API instance
   */
  public readonly api: Api;

  constructor(scope: Construct, id: string, props?: CalculateApiProps) {
    super(scope, id);

    this.api = new Api(this, id, {
      defaultAuthorizer: Authorizers.iam(),
      corsOptions: {
        allowOrigins: Cors.ALL_ORIGINS,
        allowMethods: Cors.ALL_METHODS,
      },
      integrations: {
        calculate: {
          integration: Integrations.lambda(new CalculateFunction(this,
            "CalculateFunction"))
        }
      },
      policy: new PolicyDocument({
        statements: [
          // Here we grant any AWS credentials from the account that the prototype
          is deployed in to call the api.
          // Machine to machine fine-grained access can be defined here using more
          specific principals (eg roles or
          // users) and resources (ie which api paths may be invoked by which
          principal) if required.
          // If doing so, the cognito identity pool authenticated role must still
          be granted access for cognito users to
          // still be granted access to the API.
          new PolicyStatement({
            effect: Effect.ALLOW,
            principals: [new AccountPrincipal(Stack.of(this).account)],
            actions: ["execute-api:Invoke"],
            resources: ["execute-api:/*"],
          }),
          // Open up OPTIONS to allow browsers to make unauthenticated preflight
          requests
        ]
      })
    });
  }
}
```



```
        new PolicyStatement({
            effect: Effect.ALLOW,
            principals: [new AnyPrincipal()],
            actions: ["execute-api:Invoke"],
            resources: ["execute-api:/*/OPTIONS/*"],
        }),
    ],
}),
});

// Grant authenticated users access to invoke the api
props?.userIdentity.identityPool.authenticatedRole.addToPrincipalPolicy(
    new PolicyStatement({
        effect: Effect.ALLOW,
        actions: ["execute-api:Invoke"],
        resources: [this.api.api.arnForExecuteApi("*", "/*", "*")],
    }),
);
}
```

9. Erstellen Sie die Änderungen, indem Sie den folgenden Befehl ausführen:

```
npx projen build
```

Nachdem Ihr Projekt fertig erstellt wurde, können Sie sich das aktualisierte generierte Diagramm ansehen, das Sie im `finden/packages/infra/main/cdk.out/cdkgraph/diagram.png` finden. Das Diagramm zeigt, wie die Funktion hinzugefügt und mit der erstellten API verknüpft wird. Da der CDK-Code geändert wird, wird auch dieses Diagramm aktualisiert.

Sie können Ihre Änderungen jetzt bereitstellen, indem Sie sie per Push in den Hauptzweig Ihres Repositorys übertragen und dort zusammenführen.

## Schritt 5: Codeänderungen per Push übertragen und zusammenführen

Übernehmen und übertragen Sie Ihre Codeänderungen, die dann mit dem Hauptzweig Ihres Quell-Repositorys zusammengeführt werden können.

Um Änderungen an Ihren Feature-Branch weiterzuleiten

- Führen Sie die folgenden Befehle aus, um Änderungen an Ihrem Feature-Branch festzuschreiben und per Push zu übertragen:

```
git add .
```

```
git commit -m "my commit message"
```

```
git push
```

Durch das Übertragen von Änderungen wird ein neuer Workflow-Lauf für Ihren Feature Branch ausgelöst, den Sie in der CodeCatalyst Konsole anzeigen können. Anschließend können Sie eine Pull-Anfrage erstellen, um die Änderungen mit dem Hauptzweig Ihres Quell-Repositorys zusammenzuführen. Wenn Sie den Feature-Branch mit Ihrem Haupt-Branch zusammenführen, wird der Release-Workflow ausgelöst. Du kannst den Pull-Request auch mit deinem Issue verknüpfen.

Um einen Pull-Request zu erstellen und ihn mit deinem Issue zu verknüpfen


1. Führen Sie in Ihrem Monorepo-Projekt einen der folgenden Schritte aus:
  - Wählen Sie im Navigationsbereich Code, dann Pull Requests und dann Create Pull Request aus.
  - Wählen Sie auf der Repository-Startseite die Option Mehr und dann Pull-Request erstellen aus.
  - Wähle auf der Projektseite die Option Pull-Request erstellen aus.
2. Stellen Sie im Quell-Repository sicher, dass das angegebene Quell-Repository das ist, das den festgeschriebenen Code enthält. Diese Option wird nur angezeigt, wenn Sie den Pull-Request nicht von der Hauptseite des Repositorys aus erstellt haben.
3. Wählen Sie im Zielzweig den Hauptzweig aus, in den der Code nach der Überprüfung zusammengeführt werden soll.
4. Wählen Sie im Quellzweig den Feature-Branch aus, der den festgeschriebenen Code enthält.
5. Geben Sie im Feld Titel der Pull-Anfrage einen Titel ein, der anderen Benutzern hilft zu verstehen, was überprüft werden muss und warum.
6. (Optional) Geben Sie unter Beschreibung des Pull-Requests Informationen wie einen Link zu Problemen oder eine Beschreibung Ihrer Änderungen ein.

 Tip

Du kannst Beschreibung für mich schreiben wählen, damit CodeCatalyst automatisch eine Beschreibung der in der Pull-Anfrage enthaltenen Änderungen generiert wird. Sie können Änderungen an der automatisch generierten Beschreibung vornehmen, nachdem Sie sie dem Pull Request hinzugefügt haben.

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Verwaltung generativer KI-Funktionen in Amazon CodeCatalyst](#).

- Wählen Sie unter Probleme die Option Probleme verknüpfen und wählen Sie dann das Problem aus, das Sie in erstellt haben [Schritt 2: Erstellen Sie Probleme, um zusammenzuarbeiten und die Arbeit nachzuverfolgen](#). Um die Verknüpfung eines Vorgangs aufzuheben, wählen Sie das Symbol „Verknüpfung aufheben“.
- (Optional) Wählen Sie unter Erforderliche Prüfer die Option Erforderliche Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Erforderliche Prüfer müssen die Änderungen genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.

 Note

Sie können einen Prüfer nicht sowohl als erforderlichen Prüfer als auch als optionalen Prüfer hinzufügen. Sie können sich nicht selbst als Prüfer hinzufügen.

- (Optional) Wählen Sie unter Optionale Prüfer die Option Optionale Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Optionale Prüfer müssen die Änderungen nicht als Anforderung genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.
- Ihr Pull-Request muss von den Reviewern oder Ihnen überprüft und mit dem Hauptbranch zusammengeführt werden. Weitere Informationen finden Sie unter [Eine Pull-Anfrage zusammenführen](#).

Wenn deine Änderungen mit dem Hauptzweig deines Quell-Repositorys zusammengeführt werden, wird automatisch ein neuer Workflow ausgelöst.

- Nachdem die Zusammenführung abgeschlossen ist, können Sie Ihr Problem in den Bereich Fertig verschieben.

- a. Wählen Sie im Navigationsbereich Probleme aus.
- b. Wählen Sie das Problem aus, das in erstellt wurde [Schritt 2: Erstellen Sie Probleme, um zusammenzuarbeiten und die Arbeit nachzuverfolgen](#), wählen Sie das Dropdownmenü Status und dann Fertig aus.

Der Release-Workflow stellt Ihre Anwendung nach einer erfolgreichen Ausführung bereit, sodass Sie die Änderungen einsehen können.

Um den Release-Workflow zu bestätigen und Ihre Website aufzurufen

1. Wählen Sie im Navigationsbereich Ihres Monorepo-Projekts CI/CD und dann Workflows aus.
2. Wählen Sie für den Release-Workflow den zuletzt ausgeführten Workflow aus, um die Details anzuzeigen. Weitere Informationen finden Sie unter [Anzeigen des Status und der Details einer einzelnen Ausführung](#).
3. Nachdem die Workflow-Ausführung erfolgreich abgeschlossen wurde, wählen Sie die letzte Aktion im Workflow (Deploy-B eta-ap-souteast -2) und dann Variablen aus.
4. Zeigen Sie die bereitgestellte Website an, indem Sie den Link aus der Zeile **myPDKapi** `websiteDistributionDomain NameXXXX` kopieren und in ein neues Browserfenster einfügen.
5. Geben Sie den Benutzernamen und das Passwort ein, die Sie unter erstellt haben, und wählen Sie dann Anmelden aus. [Schritt 6: Bestätigen Sie den Release-Workflow und sehen Sie sich Ihre Website an](#)
6. (Optional) Testen Sie die Änderungen in Ihrer Anwendung.
  - a. Wählen Sie das POST-Dropdown-Menü.
  - b. Geben Sie zwei Werte für `numberA` und `number B` ein und wählen Sie dann Ausführen.
  - c. Bestätigen Sie die Ergebnisse im Antworttext.

Im Laufe der Zeit können sich die Katalogversionen der PDK-Blueprints ändern. Sie können die Blueprints Ihres Projekts auf die Katalogversionen umstellen, um über die neuesten Änderungen auf dem Laufenden zu bleiben. Sie können sich die Codeänderungen und die betroffenen Umgebungen ansehen, bevor Sie die Blueprint-Versionen Ihres Projekts ändern. Weitere Informationen finden Sie unter [Aktualisierung eines Blueprints in einem Projekt](#).

# Leerzeichen in CodeCatalyst

Sie erstellen einen Bereich, der Sie, Ihr Unternehmen, Ihre Abteilung oder Gruppe repräsentiert und einen Ort bietet, an dem Ihre Entwicklungsteams Projekte verwalten können. Sie müssen einen Bereich erstellen, in dem Sie Projekte, Mitglieder und die zugehörigen Cloud-Ressourcen hinzufügen können, die Sie in Amazon erstellen CodeCatalyst.

## Note

Raumnamen müssen überall eindeutig sein CodeCatalyst. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden.

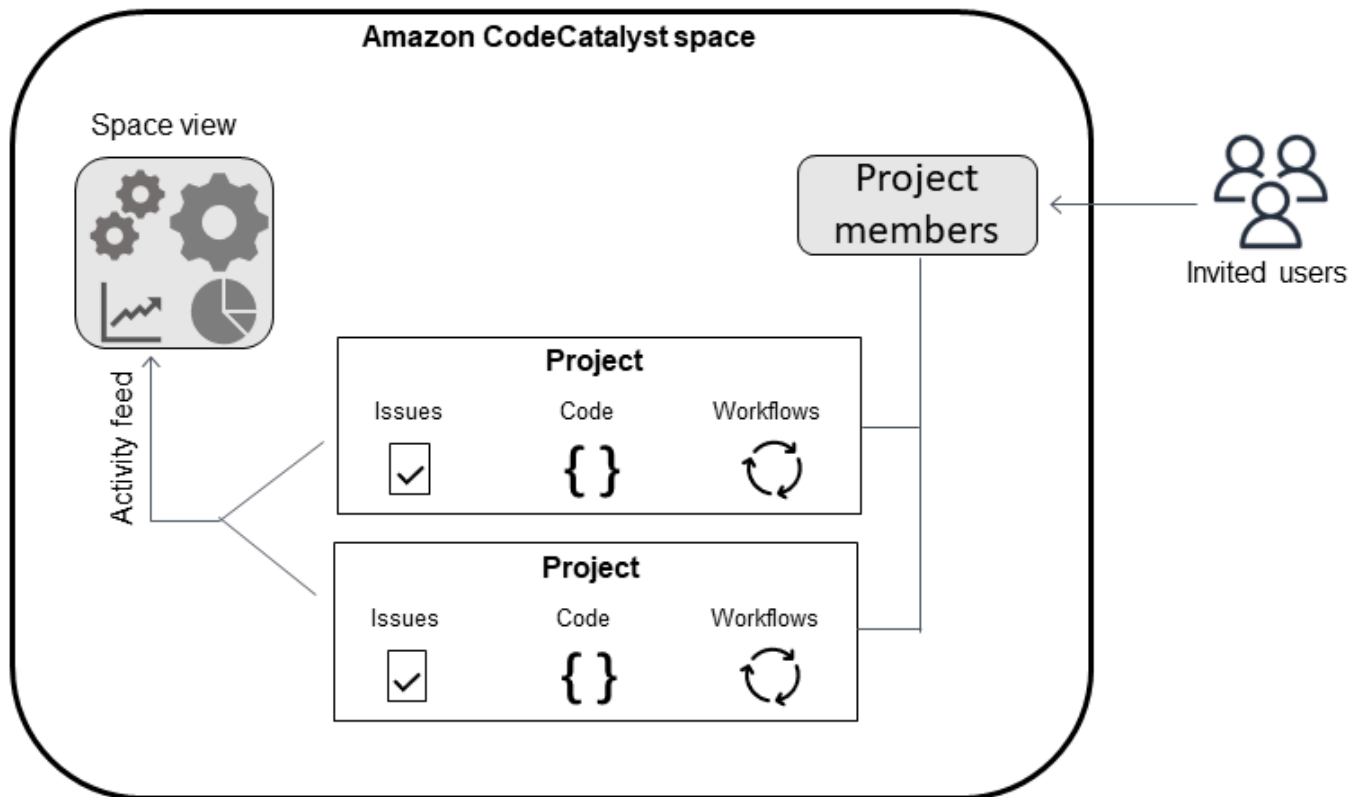
Wenn Sie einen Bereich erstellen, wird Ihnen automatisch die Rolle Space-Administrator zugewiesen. Sie können diese Rolle anderen Benutzern im Bereich hinzufügen.

Mit der Rolle Space-Administrator können Sie den Bereich wie folgt verwalten:

- Fügen Sie dem Bereich weitere Bereichsadministratoren hinzu
- Ändern Sie die Rollen und Berechtigungen der Mitglieder
- Bearbeiten oder löschen Sie den Bereich
- Erstelle Projekte und lade Mitglieder zum Projekt ein
- Sehen Sie sich eine Liste aller Projekte im Bereich an
- Sehen Sie sich den Aktivitätsfeed für alle Projekte im Bereich an

Wenn Sie einen Bereich erstellen, werden Sie dem Bereich automatisch mit zwei Rollen hinzugefügt: der Rolle des Bereichsadministrators und der Rolle des Projektadministrators für das Projekt, das Sie im Rahmen der Erstellung des Bereichs erstellt haben. Zusätzliche Benutzer werden dem Bereich automatisch als Mitglieder hinzugefügt, wenn sie Einladungen zu Projekten annehmen. Diese Mitgliedschaft im Bereich gewährt keine Berechtigungen für den Bereich. Was Benutzer in einem Bereich tun können, hängt von der Rolle ab, die der Benutzer in einem bestimmten Projekt innehat.

Weitere Informationen zu Rollen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).



Im Folgenden finden Sie weitere Überlegungen zu hinzugefügten Konten:

- Es gibt eine one-to-one Zuordnung der Kontoverbindung AWS-Konto zu einem Bereich. Ein einzelner AWS-Konto Bereich kann mehreren verschiedenen Bereichen hinzugefügt werden. AWS-Konten, für die Sie bereitstellen, müssen nicht eindeutig sein und können von mehr als einem Bereich verwendet werden.
- AWS-Konten zu einem CodeCatalyst Bereich hinzugefügt, kann in jedem Projekt in diesem Bereich verwendet werden.
- Jede Umgebung unterstützt zwar mehrere AWS-Konten, Sie können jedoch nur ein Konto pro Umgebung für eine Aktion verwenden.
- Die Abrechnung wird auf Bereichsebene konfiguriert. Für die Abrechnung können mehrere Konten konfiguriert werden, in einem CodeCatalyst Bereich kann jedoch nur eines aktiv sein. Nur eines AWS-Konto kann als Rechnungskonto für einen Bereich in verwendet werden CodeCatalyst. Wenn ein Konto bereits für einen Bereich verwendet wird, müssen Sie für den zusätzlichen Speicherplatz ein anderes Rechnungskonto verwenden.
- Nachdem Sie eine Verbindung hergestellt haben, müssen Sie Ihrer Verbindung AWS IAM-Rollen hinzufügen, wenn Ihr Workflow mit Ihrer CodeCatalyst Umgebung auf diese IAM-Rollen zugreifen

muss. Weitere Informationen zur Verwendung von Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#)

## Themen

- [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#)
- [Einen Bereich bearbeiten](#)
- [Löschen eines Leerzeichens in CodeCatalyst](#)
- [Überwachung der Aktivität für einen Bereich](#)
- [Verwaltung AWS-Konten für einen Raum](#)
- [Verwaltung von IAM-Rollen für verbundene Konten](#)
- [Verwaltung von Space-Benutzern](#)
- [Teams managen](#)
- [Verwaltung von Maschinenressourcen](#)
- [Verwaltung von Entwicklungsumgebungen für einen Bereich](#)
- [Kontingente für Leerzeichen in CodeCatalyst](#)

## Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt

Wenn Sie sich zum ersten Mal CodeCatalyst mit Ihrer AWS Builder-ID bei Amazon anmelden, müssen Sie einen Bereich erstellen. Weitere Informationen finden Sie unter [Einrichten CodeCatalyst](#). Sie können wählen, ob Sie zusätzliche Bereiche einrichten möchten, um Ihren Geschäftsanforderungen gerecht zu werden.


### Note

Bereichsnamen müssen überall eindeutig sein CodeCatalyst. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden.

Die Informationen in diesem Handbuch beziehen sich auf die Erstellung von Leerzeichen CodeCatalyst , die AWS Builder ID-Benutzer unterstützen. Die Schritte zum Einrichten und Verwalten eines Bereichs, der Identity Federation unterstützt, finden Sie im CodeCatalyst Administratorhandbuch. Informationen zur Arbeit mit Bereichen, die für den Identitätsverbund

eingrichtet sind, finden Sie unter [Einrichtung und Verwaltung von CodeCatalyst Spaces](#) im CodeCatalyst Amazon-Administratorhandbuch.

Um zusätzliche Bereiche zu erstellen, die AWS Builder ID-Benutzer unterstützen, muss Ihnen die Rolle Space-Administrator zugewiesen werden.

 Note

Wenn Sie einen zusätzlichen Bereich erstellen, werden Sie nicht aufgefordert, ein Projekt zu erstellen. Informationen zum Erstellen von Projekten in einem Bereich finden Sie unter [Ein Projekt in Amazon erstellen CodeCatalyst](#).


Um einen anderen Raum zu erstellen

1. Stellen Sie im sicher AWS Management Console, dass Sie mit dem Konto angemeldet sind AWS-Konto , das Sie Ihrem CodeCatalyst Bereich zuordnen möchten.
2. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Navigiere zu deinem Bereich.

 Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

4. Wählen Sie Bereich erstellen.
5. Geben Sie auf der Seite Bereich erstellen im Feld Bereichsname einen Namen für den Bereich ein. Sie können dies später nicht ändern.

 Note

Raumnamen müssen überall eindeutig sein CodeCatalyst. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden.

6. Wählen Sie unter die Region aus AWS-Region, in der Sie Ihre Raum- und Projektdaten speichern möchten. Sie können dies später nicht ändern.
7. Geben Sie im Feld AWS-Konto ID die zwölfstellige ID für das Konto ein, das Sie mit Ihrem Bereich verbinden möchten.



Kopieren Sie unter AWS Kontobestätigungstoken die generierte Token-ID. Das Token wird automatisch für Sie kopiert. Möglicherweise möchten Sie es jedoch speichern, während Sie die AWS Verbindungsanfrage genehmigen.

8. Wählen Sie Verifizieren in AWS.
9. Die Seite Amazon CodeCatalyst Space verifizieren wird im geöffnet AWS Management Console. Dies ist die Amazon CodeCatalyst Spaces-Seite. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

Stellen Sie sicher AWS Management Console, dass Sie dort den AWS-Region Ort auswählen, an dem Sie Ihren Bereich einrichten möchten.

Um direkt auf die Seite zuzugreifen, melden Sie sich AWS Management Console unter <https://console.aws.amazon.com/codecatalyst/home/> bei Amazon CodeCatalyst Spaces an.

Das Verifizierungstoken wird automatisch in das Bestätigungs-Token eingegeben. Ein Erfolgsbanner zeigt eine Meldung, dass es sich bei dem Token um ein gültiges Token handelt.

10. Wählen Sie Bereich verifizieren aus.

Es wird eine Erfolgsmeldung „Konto bestätigt“ angezeigt, die darauf hinweist, dass das Konto dem Bereich hinzugefügt wurde.

11. Bleiben Sie auf der Seite Amazon CodeCatalyst Space verifizieren. Wählen Sie den folgenden Link: Um IAM-Rollen für diesen Bereich hinzuzufügen, sehen Sie sich die Bereichsdetails an.

Die Seite mit den CodeCatalyst Space-Details wird im AWS Management Console geöffnet. Dies ist die Amazon CodeCatalyst Spaces-Seite. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

12. Wählen Sie unter IAM-Rollen verfügbar für die CodeCatalyst Option IAM-Rolle hinzufügen aus.

Die Seite Verfügbare IAM-Rollen hinzufügen wird angezeigt. CodeCatalyst

13. Wählen Sie in IAM die Option CodeCatalyst Entwicklungsadministratorrolle erstellen aus. Mit dieser Option wird eine Servicerolle erstellt, die die Berechtigungsrichtlinie und die Vertrauensrichtlinie für die Entwicklungsrolle enthält.

Die Entwicklerrolle ist eine AWS IAM-Rolle, die es Ihren CodeCatalyst Workflows ermöglicht, auf AWS Ressourcen wie Amazon S3, Lambda und zuzugreifen. AWS CloudFormation Der Rolle wird ein Name `CodeCatalystWorkflowDevelopmentRole-spaceName` mit einer angehängten eindeutigen Kennung zugewiesen. Weitere Informationen

zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#).

14. Wählen Sie Entwicklungsrolle erstellen aus.
15. Sehen Sie sich auf der Verbindungsseite unter Verfügbare IAM-Rollen die Entwicklerrolle in der Liste der IAM-Rollen an, die Ihrem Konto hinzugefügt wurden. CodeCatalyst
16. Wählen Sie Gehe zu Amazon CodeCatalyst.
17. Wählen Sie auf der Erstellungsseite in CodeCatalyst die Option Bereich erstellen aus.

## Einen Bereich bearbeiten

Sie können die Beschreibung eines Bereichs ändern, damit Benutzer besser verstehen, wofür er gedacht ist.

Sie benötigen die Rolle eines Space-Administrators, um Space-Details bearbeiten zu können.

Die Informationen in diesem Handbuch beziehen sich auf die Bearbeitung von Bereichen CodeCatalyst , die AWS Builder ID-Benutzer unterstützen. Weitere Informationen zu den Schritten zur Einrichtung und Verwaltung eines Bereichs, der Identity Federation unterstützt, finden Sie unter [Einrichtung und Verwaltung von CodeCatalyst Spaces](#) im CodeCatalyst Amazon-Administratorhandbuch.

Um eine Bereichsbeschreibung zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie auf der Registerkarte Bereichseinstellungen die Option Bearbeiten aus. Nehmen Sie die gewünschten Änderungen an der Bereichsbeschreibung vor und wählen Sie dann Speichern.

## Löschen eines Leerzeichens in CodeCatalyst

Sie können einen Bereich löschen, um den Zugriff auf alle Ressourcen des Bereichs zu entziehen. Sie benötigen die Rolle eines Space-Administrators, um einen Bereich löschen zu können.

### Note

Das Löschen eines Bereichs kann nicht rückgängig gemacht werden.

Nachdem Sie einen Bereich gelöscht haben, können alle Space-Mitglieder nicht mehr auf die Space-Ressourcen zugreifen. Die Abrechnung von Speicherressourcen wird ebenfalls eingestellt, und alle Workflows, die von Quell-Repositories von Drittanbietern veranlasst werden, werden gestoppt.

### Note

Raumnamen müssen überall CodeCatalyst eindeutig sein. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden.

Die Informationen in diesem Handbuch beziehen sich auf das Löschen von Leerzeichen in CodeCatalyst Dateien, die AWS Builder ID-Benutzer unterstützen. Weitere Informationen zu den Schritten zur Einrichtung und Verwaltung eines Bereichs, der Identity Federation unterstützt, finden Sie unter [Einrichtung und Verwaltung von CodeCatalyst Spaces](#) im CodeCatalyst Amazon-Administratorhandbuch.

Um einen Bereich zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie „Einstellungen“ und anschließend „Löschen“.
4. Geben Sie **delete** ein, um den Löschvorgang zu bestätigen.

## 5. Wählen Sie Löschen aus.

### Note

Wenn Sie zu mehr als einem Bereich gehören, werden Sie auf die Übersichtsseite des Bereichs weitergeleitet. Wenn du zu einem Space gehörst, wirst du auf die Seite zur Erstellung eines Bereichs weitergeleitet.

## Überwachung der Aktivität für einen Bereich

Um kürzlich erstellte Projekte und Statusmeldungen zu sehen, können Sie in der CodeCatalyst Konsole einen Aktivitätsfeed aufrufen, in dem Aktualisierungen für Space-Ressourcen angezeigt werden.

Im Aktivitätsfeed können Sie Messwerte wie fehlgeschlagene Workflow-Ausführungen und erstellte Projekte einsehen.

Um die Aktivitäten in deinem Bereich zu sehen

1. Öffne die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Activity (Aktivität).
4. Sehen Sie sich die Informationen unter Aktivität an.
5. Um nach Aktivität zu filtern, wählen Sie den Selektor oben rechts aus.
6. Um alle Aktivitäten in deinem Bereich zu sehen, wähle Beliebiger Aktivitätstyp aus.

## Verwaltung AWS-Konten für einen Raum

Sie können Ressourcen aus Ihren AWS-Konten Amazon CodeCatalyst Spaces verwenden. Dazu müssen Sie eine Verbindung zwischen dem AWS-Konten und Ihrem Bereich einrichten

CodeCatalyst. Wenn Sie eine solche Verbindung herstellen, können Projekte und Workflows in Ihrem CodeCatalyst Bereich mit Ressourcen in Ihrem Bereich interagieren AWS-Konten. Sie müssen für jede Verbindung, die AWS-Konto Sie mit Ihrem CodeCatalyst Space verwenden möchten, eine Verbindung herstellen.

Nachdem Sie eine Verbindung hergestellt haben, können Sie ihr AWS IAM-Rollen zuordnen.

## Themen

- [Ein AWS-Konto zu einem Leerzeichen hinzufügen](#)
- [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#)
- [Hinzufügen der Kontoverbindung und der IAM-Rollen zu Ihrer Bereitstellungsumgebung](#)
- [Kontoverbindungen anzeigen](#)
- [Konto aus einem Bereich entfernen \(in CodeCatalyst\)](#)

Sie können die Nutzung CodeCatalyst mit Autorisierung einrichten, AWS-Konten indem Sie die Konten zu Ihrem Bereich hinzufügen. Durch AWS-Konten das Hinzufügen zu Ihrem CodeCatalyst Bereich können Sie Ihren Projekt-Workflows Zugriff auf AWS-Konto Ressourcen und Ihre Abrechnungskonfiguration gewähren.

Durch das Hinzufügen AWS-Konto eines wird eine Verbindung hergestellt, die CodeCatalyst zur Nutzung dieses Kontos berechtigt. Sie können added verwenden AWS-Konten , um Folgendes zu tun:

- Richten Sie die Abrechnung für einen CodeCatalyst Raum ein. Weitere Informationen finden Sie unter [Abrechnung verwalten](#) im CodeCatalyst Amazon-Administratorhandbuch.
- Erlauben CodeCatalyst Sie die Übernahme von IAM-Rollen für den Zugriff auf AWS Ressourcen und deren Bereitstellung AWS-Services im Konto. Siehe [Verwaltung AWS-Konten für einen Raum](#).

Kontoverbindungen werden hergestellt, indem die Autorisierung mit dem AWS-Konto abgeschlossen wird. Nachdem die Verbindung hergestellt wurde, konfigurieren Sie die Verbindung weiter für Workflows und Projekte, die verwendet werden sollen, indem Sie IAM-Rollen hinzufügen.

## Ein AWS-Konto zu einem Leerzeichen hinzufügen

Sie verwenden die CodeCatalyst Konsole und die AWS Management Console , um Ihren Bereich mit einem zu verbinden AWS-Konto.

Bevor Sie einem Bereich in eine AWS-Konto hinzufügen CodeCatalyst, müssen Sie die folgenden Voraussetzungen erfüllen:

- Erstellen Sie eine AWS IAM-Rollen in dem Konto, mit dem Sie eine Verbindung herstellen möchten, AWS-Konto und erwerben Sie Berechtigungen zum Erstellen von IAM-Rollen.
- Erstellen Sie die IAM-Rolle (n), die Sie Ihrer Kontoverbindung zuordnen möchten, einschließlich der IAM-Richtlinien mit den Berechtigungen für die Rollen.
- Erwerben Sie die Rolle eines Space-Administrators in dem CodeCatalyst Bereich, in dem Sie die Verbindung herstellen möchten.

## Themen

- [Schritt 1: Verbindungsanfrage erstellen](#)
- [Schritt 2: Annahme einer Kontoverbindungsanfrage](#)
- [Schritt 3: Überprüfen Sie eine genehmigte Verbindung](#)
- [Schritt 4: Fügen Sie Ihrer Verbindung IAM-Rollen hinzu](#)
- [Nächste Schritte: Erstellen Sie zusätzliche IAM-Rollen für Ihre Kontoverbindung](#)

## Schritt 1: Verbindungsanfrage erstellen

Wenn Sie eine Verbindungsanforderung in der CodeCatalyst Konsole erstellen, wird ein Verbindungstoken generiert, mit dem Sie die Autorisierung abschließen können.

Sie müssen in dem Bereich, in dem Sie die Verbindung herstellen möchten, über die Rolle CodeCatalyst Space-Administrator oder Poweruser verfügen. Sie müssen außerdem über Administratorrechte für den Benutzer verfügen, den AWS-Konto Sie hinzufügen möchten.

So stellen Sie eine Verbindung her

1. Stellen Sie im sicher AWS Management Console, dass Sie mit demselben Konto angemeldet sind, mit dem Sie eine Verbindung herstellen möchten.
2. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
4. Wählen Sie Hinzufügen AWS-Konto.

5. Geben Sie auf der CodeCatalyst Seite **AWS-Konto Mit Amazon verbinden** im Feld **AWS-Konto ID** die zwölfstellige ID für das Konto ein, das Sie mit Ihrem Bereich verbinden möchten. Informationen zum Auffinden Ihrer AWS-Konto ID finden Sie unter [Ihre AWS-Konto ID und ihr Alias](#).
6. Geben Sie unter **CodeCatalyst Amazon-Anzeigename** einen Referenznamen für das Konto ein.
7. (Optional) Geben Sie unter **Verbindungsbeschreibung** eine Beschreibung für das Konto ein, anhand derer Sie die Projekte auswählen können, für die das Konto und die Rolle (n) gelten.
8. Wählen Sie **Associate AWS-Konto** aus.
9. Die Seite kehrt zur **AWS-Konto Detailseite** zurück, auf der ein Erfolgsbanner angezeigt wird.

## Schritt 2: Annahme einer Kontoverbindungsanfrage

Nachdem Sie in der CodeCatalyst Konsole eine Verbindungsanfrage eingereicht haben, stimmen Sie sich mit Ihrem AWS Administrator ab AWS-Konto, um die Verbindungsanfrage zu akzeptieren, indem Sie sie mit dem bereitgestellten Verbindungstoken einreichen.

Stellen Sie sicher, dass Sie über Administratorrechte für Ihr Konto verfügen und dass Sie AWS Management Console mit dem Konto angemeldet sind, AWS-Konto für das Sie die Verbindung herstellen.


Um eine Verbindungsanfrage zu genehmigen (Konsole)

1. Stellen Sie in der sicher AWS Management Console, dass Sie mit demselben Konto angemeldet sind, mit dem Sie eine Verbindung herstellen möchten.
2. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie **Settings (Einstellungen)** und anschließend **AWS-Konten** aus.
4. Wählen Sie auf der **AWS-Konto Detailseite** unter **Vollständige Einrichtung aus AWS Management Console**.
5. Die Seite **Amazon CodeCatalyst Space verifizieren** wird im geöffnet AWS Management Console. Dies ist die **Amazon CodeCatalyst Spaces-Seite**. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

Um direkt auf die Seite zuzugreifen, melden Sie sich AWS Management Console unter <https://console.aws.amazon.com/codecatalyst/home/> bei **Amazon CodeCatalyst Spaces** an.

Das Verifizierungstoken wird automatisch in das Bestätigungs-Token eingegeben. Eine Erfolgsmeldung zeigt eine Meldung an, dass es sich bei dem Token um ein gültiges Token handelt.

6. (Optional) Wählen Sie unter Autorisierte kostenpflichtige Tarife die Option Bezahlte Tarife autorisieren (Standard, Enterprise) aus, um die kostenpflichtigen Stufen für Ihr Rechnungskonto zu aktivieren.

 Note

Dadurch wird die Abrechnungsstufe nicht auf eine kostenpflichtige Stufe hochgestuft. Dadurch wird das jedoch AWS-Konto so konfiguriert, dass Sie die Abrechnungsstufe für Ihren Speicherplatz jederzeit in CodeCatalyst ändern können. Sie können die kostenpflichtigen Stufen jederzeit aktivieren. Ohne diese Änderung kann der Speicherplatz nur das kostenlose Kontingent nutzen.

7. Wählen Sie Speicherplatz verifizieren aus.

Es wird eine Erfolgsmeldung „Konto bestätigt“ angezeigt, die darauf hinweist, dass das Konto dem Bereich hinzugefügt wurde.

### Schritt 3: Überprüfen Sie eine genehmigte Verbindung

Nachdem eine Verbindung genehmigt wurde, können Sie die Verbindung zusammen mit den IAM-Rollen, die Sie ihr hinzugefügt haben, in der Konsole anzeigen.

Um eine genehmigte Verbindung zu überprüfen

1. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
2. Die Kontoverbindung wird mit dem Datum aufgeführt, an dem sie erstellt wurde.
3. Wählen Sie den Anzeigenamen des Kontos. Die AWS-Konto Detailseite wird angezeigt.

### Schritt 4: Fügen Sie Ihrer Verbindung IAM-Rollen hinzu

Wenn Sie eine IAM-Rolle verwenden, die für eine CodeCatalyst Bereitstellungsaktion konfiguriert ist, fügen Sie die Rolle zu Ihrer Bereitstellungsumgebung hinzu. Weitere Informationen finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).



## Nächste Schritte: Erstellen Sie zusätzliche IAM-Rollen für Ihre Kontoverbindung

Nachdem Sie eine Verbindung hergestellt haben, können Sie zusätzliche IAM-Rollen erstellen, um sie zu erweitern. Die IAM-Rollen, die Sie hinzufügen, hängen von Ihren Workflows ab. Für eine CodeCatalyst Build-Aktion ist beispielsweise die CodeCatalyst Build-Rolle erforderlich.

Um Ihr Konto zu verbinden, benötigen Sie den Amazon-Ressourcennamen (ARN) für die Rollen, die Sie erstellt haben. Kopieren Sie den ARN für Ihre Rolle oder Rollen, wie hier beschrieben. Weitere Informationen zur Arbeit mit ARNs für IAM-Rollen finden Sie unter [Amazon Resource Name \(ARN\)](#).

So greifen Sie auf Ihre IAM-Rolle zu: ARN

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie hinzufügen möchten.
4. Wählen Sie die Rolle aus der Liste aus.

Die Übersichtsseite der Rolle wird angezeigt.

5. Kopieren Sie oben den ARN-Wert der Rolle.

## Hinzufügen von IAM-Rollen zu Kontoverbindungen

Zum Erstellen Ihrer Kontoverbindung gehört auch das Hinzufügen der IAM-Rolle (n), die Sie für Projekte in Ihrem CodeCatalyst Bereich verwenden möchten.

### Note

Um IAM-Rollen mit einer Kontoverbindung zu verwenden, stellen Sie sicher, dass die Vertrauensrichtlinie aktualisiert wurde, sodass sie den CodeCatalyst Dienstprinzipal verwendet.

Fügen Sie einer Kontoverbindung (Konsole) IAM-Rollen hinzu

1. Stellen Sie in der sicher AWS Management Console, dass Sie mit demselben Konto angemeldet sind, das Sie verwalten möchten.
2. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

3. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
4. Wählen Sie den CodeCatalyst Amazon-Anzeigenamen Ihrer Kontoverbindung aus und wählen Sie dann Rollen verwalten von AWS Management Console.

Die Seite „IAM-Rolle zu Amazon CodeCatalyst Space hinzufügen“ wird angezeigt.

5. Führen Sie eine der folgenden Aktionen aus:
  - Um eine Servicerolle zu erstellen, die die Berechtigungsrichtlinie und die Vertrauensrichtlinie für die Entwicklerrolle enthält, wählen Sie Create CodeCatalyst development administrator role in IAM. Der Rolle wird ein Name `CodeCatalystWorkflowDevelopmentRole-spaceName` mit einer angehängten eindeutigen Kennung zugewiesen. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#).

Wählen Sie Entwicklungsrolle erstellen aus.

- Um eine Rolle hinzuzufügen, die Sie bereits in IAM erstellt haben, wählen Sie Bestehende IAM-Rolle hinzufügen aus. Wählen Sie unter Vorhandene IAM-Rolle auswählen die Rolle aus der Dropdownliste aus.

Wählen Sie Rolle hinzufügen aus.

Die Seite wird in der AWS Management Console geöffnet. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

6. Wählen Sie im Navigationsbereich der Amazon CodeCatalyst Spaces-Seite die Option Spaces aus.


Um direkt auf die Seite zuzugreifen, melden Sie sich AWS Management Console unter <https://console.aws.amazon.com/codecatalyst/home/> bei Amazon CodeCatalyst Spaces an.

7. Wählen Sie das Konto aus, das für Ihren CodeCatalyst Bereich hinzugefügt wurde. Die Verbindungsseite wird angezeigt.
8. Sehen Sie sich auf der Verbindungsseite unter Verfügbare IAM-Rollen die Liste der IAM-Rollen an CodeCatalyst, die Ihrem Konto hinzugefügt wurden. Wählen Sie IAM-Rolle zuordnen mit aus. CodeCatalyst

9. Geben Sie im Pop-up Eine IAM-Rolle zuordnen unter Rollen-ARN den Amazon-Ressourcennamen (ARN) der IAM-Rolle ein, die Sie Ihrem CodeCatalyst Bereich zuordnen möchten.

Wählen Sie unter Zweck einen Zweck der Rolle aus, der beschreibt, wie Sie die Rolle in Ihrer Kontoverbindung verwenden möchten. Geben Sie RUNNER die Rollen an, die Sie zum Ausführen von Aktionen in Workflows verwenden. Geben Sie SERVICE die Rollen an, die Sie für den Zugriff auf einen anderen Dienst verwenden.

Sie können mehr als einen Zweck angeben.

 Note

Die Wahl eines Zwecks für die Rolle ARN ist erforderlich.

10. Wählen Sie Eine IAM-Rolle zuordnen aus. Wiederholen Sie diese Schritte für weitere IAM-Rollen.

## Hinzufügen der Kontoverbindung und der IAM-Rollen zu Ihrer Bereitstellungsumgebung

Um auf AWS Ressourcen wie Amazon ECS oder AWS Lambda Ressourcen für Bereitstellungen zugreifen zu können, benötigen CodeCatalyst Build- und Deployment-Aktionen IAM-Rollen mit Zugriffsberechtigungen für diese Ressourcen. Mit der Rolle Space-Administrator oder Poweruser können Sie Ihr CodeCatalyst Konto mit dem Ort verbinden, an AWS-Konto dem Ihre Ressourcen erstellt wurden. Anschließend fügen Sie die IAM-Rolle zu Ihrer Kontoverbindung hinzu. Für Bereitstellungsaktionen müssen Sie dann die IAM-Rolle zu einer CodeCatalyst Umgebung hinzufügen.

Sie müssen die IAM-Rollen hinzufügen, die Sie mit Bereitstellungsumgebungen in Ihren Projekten verwenden möchten. Durch das Hinzufügen der Rollen zur Kontoverbindung werden die Rollen und die Verbindung zu den Bereitstellungsumgebungen des Projekts nicht hinzugefügt. Um Ihre Kontoverbindung und Ihre IAM-Rollen zu Ihrer Bereitstellungsumgebung hinzuzufügen, stellen Sie sicher, dass die Kontoverbindung und die Rollen wie unter beschrieben erstellt wurden. [Schritt 4: Fügen Sie Ihrer Verbindung IAM-Rollen hinzu](#)

Verwenden Sie dann die Seite Umgebungen in der CodeCatalyst Konsole, um Ihre Kontoverbindung und Ihre IAM-Rolle zu einer Bereitstellungsumgebung in einem Projekt hinzuzufügen.

**Note**

Sie fügen einer Umgebung nur dann eine IAM-Rolle hinzu, wenn die IAM-Rolle für eine CodeCatalyst Aktion verwendet wird, für die eine IAM-Rolle erforderlich ist. Alle Workflow-Aktionen, die IAM-Rollen erfordern, einschließlich Build-Aktionen, müssen eine Umgebung verwenden. CodeCatalyst

Um Ihre Kontoverbindung und Ihre IAM-Rollen zu Ihrer Bereitstellungsumgebung hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt mit der Bereitstellungsumgebung, in der Sie die Kontoverbindung und die IAM-Rollen hinzufügen möchten.
3. Erweitern Sie CI/CD und wählen Sie dann Umgebungen aus.
4. Wählen Sie Ihre Umgebung aus, und dann werden die zusätzlichen Registerkarten angezeigt.
5. Wählen Sie die Registerkarte AWS-Konto Verbindungen. Unter Verbindungsname werden die Konten aufgeführt, die der Umgebung hinzugefügt wurden, falls vorhanden.
6. Wählen Sie Associate AWS-Konto aus. Die <environment\_name>Seite „Zuordnen AWS-Konto mit“ wird angezeigt.
7. Wählen Sie unter Verbindung den Namen der Kontoverbindung mit den IAM-Rollen aus, die Sie hinzufügen möchten. Wählen Sie Associate aus.

## Kontoverbindungen anzeigen

Sie können eine Liste Ihrer Verbindungen und Details zu jeder Verbindung einsehen.

Sie benötigen die Rolle Space-Administrator oder Poweruser, um Verbindungen für Ihren Space verwalten zu können.

Um alle Verbindungen für einen CodeCatalyst Space anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit der Kontoverbindung, die Sie anzeigen möchten.
3. Wählen Sie den Tab „AWS Konten“.
4. Sehen Sie sich unter AWS Konten die Liste der Kontoverbindungen für den Bereich an, einschließlich der Konto-ID und des Status für jede Verbindung.

So zeigen Sie die Verbindungsdetails zu den Konten an

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
3. Wählen Sie Amazon CodeCatalyst Amazon-Anzeigenname den Verbindungsnamen aus. Sehen Sie sich auf der Detailseite die Liste der IAM-Rollen an, die mit der Verbindung verknüpft sind, zusammen mit anderen Details.

## Konto aus einem Bereich entfernen (in CodeCatalyst)

Sie können eine Kontoverbindung löschen, die Sie nicht mehr benötigen. Bei diesem Verfahren löschen Sie CodeCatalyst eine Kontoverbindung, die Sie zuvor zu Ihrem Bereich hinzugefügt haben. Dadurch wird die Kontoverbindung aus Ihrem Bereich gelöscht, sofern es sich bei dem Konto nicht um das Rechnungskonto für den Bereich handelt.

### Important

Nachdem eine Kontoverbindung gelöscht wurde, kannst du sie nicht erneut verbinden. Sie müssen eine neue Kontoverbindung erstellen und dann je nach Bedarf IAM-Rollen und -Umgebungen zuordnen oder die Abrechnung einrichten.

Für Ihren CodeCatalyst Speicherplatz muss ein Rechnungskonto eingerichtet werden, auch wenn die Nutzung des Speicherplatzes das kostenlose Kontingent nicht überschreitet. Bevor Sie einen Speicherplatz für ein Konto entfernen können, bei dem es sich um ein bestimmtes Rechnungskonto handelt, müssen Sie ein weiteres Konto für Ihren Speicherplatz hinzufügen. Weitere Informationen finden Sie unter [Abrechnung verwalten](#) im CodeCatalyst Amazon-Administratorhandbuch.

### Important

Sie können zwar diese Schritte verwenden, um ein Konto zu entfernen, dies wird jedoch nicht empfohlen. Das Konto kann auch zur Unterstützung von Workflows in eingerichtet werden CodeCatalyst.

Um die Kontoverbindungen für Ihren Space verwalten zu können, benötigen Sie die Rolle Space-Administrator oder Poweruser.

Um eine Kontoverbindung zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
3. Wählen Sie unter CodeCatalyst Amazon-Anzeigename die Auswahl neben der Kontoverbindung aus, die Sie entfernen möchten.
4. Wählen Sie Remove (Entfernen) AWS-Konto aus. Bestätigen Sie den Löschvorgang, indem Sie den Namen in das Feld eingeben und dann Entfernen wählen.

Ein Erfolgsbanner wird angezeigt und die Kontoverbindung wird aus der Verbindungsliste entfernt.

## Verwaltung von IAM-Rollen für verbundene Konten

Sie erstellen Rollen in AWS Identity and Access Management (IAM) für das Konto, dem Sie etwas hinzufügen möchten. CodeCatalyst Wenn Sie ein Rechnungskonto hinzufügen, müssen Sie keine Rollen erstellen.

In Ihrem müssen Sie über die Berechtigungen verfügen AWS-Konto, um Rollen für die Rollen zu erstellen, die AWS-Konto Sie Ihrem Bereich hinzufügen möchten. Weitere Informationen zu IAM-Rollen und -Richtlinien, einschließlich IAM-Referenzen und Beispielrichtlinien, finden Sie unter [Identity and Access Management und Amazon CodeCatalyst](#) Weitere Informationen zu den in verwendeten Vertrauensrichtlinien und Dienstprinzipalen finden Sie unter CodeCatalyst. [CodeCatalyst Vertrauensmodell](#)

In CodeCatalyst müssen Sie mit der Rolle Space-Administrator angemeldet sein, um die Schritte zum Hinzufügen von Konten (und der Rollen, falls zutreffend) zu Ihrem Space ausführen zu können.

Sie können Ihren Kontoverbindungen Rollen hinzufügen, indem Sie eine der folgenden Methoden verwenden.

- Informationen zum Erstellen einer Servicerolle, die die Berechtigungsrichtlinie und die Vertrauensrichtlinie für die CodeCatalystWorkflowDevelopmentRole-*spaceName*Rolle enthält, finden Sie unter [CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle](#).

- Ein Beispiel für das Erstellen einer Rolle und das Hinzufügen einer Richtlinie zum Erstellen eines Projekts anhand eines Blueprints finden Sie unter [Eine IAM-Rolle erstellen und die CodeCatalyst Vertrauensrichtlinie verwenden](#).
- Eine Liste mit Beispiel-Rollenrichtlinien, die Sie beim Erstellen Ihrer IAM-Rollen verwenden können, finden Sie unter. [IAM-Rollen für Amazon- CodeCatalyst Zugriff auf -AWSRessourcen](#)
- Eine ausführliche Anleitung zum Erstellen von Rollen für Workflow-Aktionen finden Sie im Workflow-Tutorial für diese Aktion wie folgt:
  - [Tutorial: Artefakte auf Amazon S3 hochladen](#)
  - [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#)
  - [Tutorial: Bereitstellen einer Anwendung in Amazon ECS](#)
  - [Tutorial: Lint-Code mit einer GitHub Aktion](#)

## Themen

- [CodeCatalystWorkflowDevelopmentRole-spaceName Rolle](#)
- [AWSRoleForCodeCatalystSupport Rolle](#)
- [Eine IAM-Rolle erstellen und die CodeCatalyst Vertrauensrichtlinie verwenden](#)

## CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle

Sie erstellen die Entwicklerrolle als 1-Klick-Rolle in IAM. Sie müssen in dem Bereich, in dem Sie das Konto hinzufügen möchten, über die Rolle Space-Administrator oder Poweruser verfügen. Sie müssen außerdem über Administratorrechte für das Konto verfügen, das AWS-Konto Sie hinzufügen möchten.

Bevor Sie mit dem unten beschriebenen Verfahren beginnen, müssen Sie sich AWS Management Console mit demselben Konto, das Sie Ihrem CodeCatalyst Bereich hinzufügen möchten, bei dem anmelden. Andernfalls gibt die Konsole einen unbekanntem Kontofehler zurück.

Um das zu erstellen und hinzuzufügen CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die AWS Management Console, und stellen Sie dann sicher, dass Sie mit derselben AWS-Konto für Ihren Bereich angemeldet sind.
2. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

3. Navigiere zu deinem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
4. Wählen Sie den Link für den AWS-Konto Ort aus, an dem Sie die Rolle erstellen möchten. Die AWS-Konto Detailseite wird angezeigt.
5. Wählen Sie Rollen verwalten von AWS Management Console.

Die Seite „IAM-Rolle zu Amazon CodeCatalyst Space hinzufügen“ wird in der AWS Management Console geöffnet. Dies ist die Amazon CodeCatalyst Spaces-Seite. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

6. Wählen Sie in IAM die Option CodeCatalyst Entwicklungsadministratorrolle erstellen aus. Mit dieser Option wird eine Servicerolle erstellt, die die Berechtigungsrichtlinie und die Vertrauensrichtlinie für die Entwicklungsrolle enthält. Die Rolle wird einen Namen haben `CodeCatalystWorkflowDevelopmentRole-spaceName`. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#).

#### Note

Diese Rolle wird nur für die Verwendung mit Entwicklerkonten empfohlen. Sie verwendet die `AdministratorAccess` AWS verwaltete Richtlinie, sodass sie vollen Zugriff hat, um darin neue Richtlinien und Ressourcen zu erstellen AWS-Konto.

7. Wählen Sie Entwicklungsrolle erstellen aus.
8. Sehen Sie sich auf der Verbindungsseite unter Verfügbare IAM-Rollen die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle in der Liste der IAM-Rollen an CodeCatalyst, die Ihrem Konto hinzugefügt wurden.
9. Um zu Ihrem Bereich zurückzukehren, wählen Sie Gehe zu Amazon CodeCatalyst.

## AWSRoleForCodeCatalystSupport Rolle

Sie erstellen die Support-Rolle als 1-Klick-Rolle in IAM. Sie müssen in dem Bereich, in dem Sie das Konto hinzufügen möchten, über die Rolle Space-Administrator oder Poweruser verfügen. Sie müssen außerdem über Administratorrechte für das Konto verfügen, das AWS-Konto Sie hinzufügen möchten.



Bevor Sie mit dem unten beschriebenen Verfahren beginnen, müssen Sie sich AWS Management Console mit demselben Konto, das Sie Ihrem CodeCatalyst Bereich hinzufügen möchten, bei dem anmelden. Andernfalls gibt die Konsole einen unbekanntem Kontofehler zurück.

Um das zu erstellen und hinzuzufügen CodeCatalyst AWSRoleForCodeCatalystSupport

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die AWS Management Console, und stellen Sie dann sicher, dass Sie mit demselben AWS-Konto für Ihren Bereich angemeldet sind.
2. Navigieren Sie zu Ihrem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
3. Wählen Sie den Link für den AWS-Konto Ort aus, an dem Sie die Rolle erstellen möchten. Die AWS-Konto Detailseite wird angezeigt.
4. Wählen Sie Rollen verwalten von AWS Management Console.

Die Seite „IAM-Rolle zu Amazon CodeCatalyst Space hinzufügen“ wird in der AWS Management Console geöffnet. Dies ist die Amazon CodeCatalyst Spaces-Seite. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

5. Wählen Sie unter CodeCatalyst Bereichsdetails die Option CodeCatalyst Support-Rolle hinzufügen aus. Mit dieser Option wird eine Servicerolle erstellt, die die Berechtigungsrichtlinie und die Vertrauensrichtlinie für die Vorschau-Entwicklungsrolle enthält. Der Rolle wird ein Name AWSRoleForCodeCatalystSupport mit einer angehängten eindeutigen Kennung zugewiesen. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur AWSRoleForCodeCatalystSupport Servicerolle](#).
6. Lassen Sie auf der Seite Rolle für CodeCatalyst Support hinzufügen die Standardeinstellung aktiviert und wählen Sie dann Rolle erstellen aus.
7. Sehen Sie sich unter Verfügbare IAM-Rollen die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle in der Liste der IAM-Rollen an, die Ihrem Konto hinzugefügt wurden. CodeCatalyst
8. Um zu Ihrem Bereich zurückzukehren, wählen Sie Gehe zu Amazon CodeCatalyst.

## Eine IAM-Rolle erstellen und die CodeCatalyst Vertrauensrichtlinie verwenden

IAM-Rollen, die CodeCatalyst mit AWS-Konto Verbindungen verwendet werden sollen, müssen so konfiguriert sein, dass sie die hier angegebene Vertrauensrichtlinie verwenden. Gehen Sie wie

folgt vor, um eine IAM-Rolle zu erstellen und eine Richtlinie anzuhängen, mit der Sie Projekte aus Blueprints in erstellen können. CodeCatalyst

Als Alternative können Sie eine Servicerolle erstellen, die die Berechtigungsrichtlinie und die Vertrauensrichtlinie für die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle enthält. Weitere Informationen finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie Roles (Rollen) und anschließend Create role (Rolle erstellen).
3. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie.
4. Fügen Sie im Formular Benutzerdefinierte Vertrauensrichtlinie die folgende Vertrauensrichtlinie ein.

```
"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/
*"
        }
      }
    }
  ]
```

5. Wählen Sie Weiter aus.
6. Suchen Sie unter Berechtigungen hinzufügen nach einer benutzerdefinierten Richtlinie, die Sie bereits in IAM erstellt haben, und wählen Sie sie aus.
7. Wählen Sie Weiter aus.

8. Geben Sie unter Rollenname einen Namen für die Rolle ein, zum Beispiel: `codecatalyst-project-role`
9. Wählen Sie Rolle erstellen aus.
10. Kopieren Sie die Rolle Amazon Resource Name (ARN). Sie müssen diese Informationen angeben, wenn Sie die Rolle zu Ihrer Kontoverbindung oder Umgebung hinzufügen.

## Verwaltung von Space-Benutzern

Sie können Mitglieder für einen Bereich verwalten, indem Sie die Rollen von Benutzern, die dem Bereich beitreten, anzeigen, hinzufügen, entfernen oder ändern.

Die Informationen in diesem Handbuch beziehen sich auf das Einladen und Verwalten von Benutzern in Bereichen CodeCatalyst, die AWS Builder ID-Benutzer unterstützen. Weitere Informationen zu den Schritten zur Einrichtung und Verwaltung eines Bereichs, der Identity Federation unterstützt, finden Sie unter [Einrichtung und Verwaltung von CodeCatalyst Spaces](#) im CodeCatalyst Amazon-Administratorhandbuch.

### Mitglieder in einem Bereich anzeigen

Sie können die Benutzer in Ihrem Bereich anzeigen, einschließlich Informationen zu ihren Anzeigenamen, Aliasnamen und der Rolle, die sie für den Bereich haben. Es gibt drei Rollen für Mitglieder in einem Bereich:

- **Bereichsadministrator** — Diese Rolle hat alle Rechte CodeCatalyst, einschließlich der Erstellung von Projekten. Weisen Sie diese Rolle nur Benutzern zu, die jeden Aspekt eines Bereichs verwalten müssen, z. B. den Zugriff auf alle Projekte im Bereich.

Sie können diese Rolle später nicht ändern, ohne den Benutzer zuerst zu entfernen. Weitere Informationen finden Sie unter [Rolle „Space-Administrator“](#).

- **Poweruser** — Diese Rolle ist die zweitstärkste Rolle in Amazon CodeCatalyst Spaces, hat aber keinen Zugriff auf Projekte in einem Bereich. Sie ist für Benutzer konzipiert, die in der Lage sein müssen, Projekte in einem Bereich zu erstellen und sie bei der Verwaltung der Benutzer und Ressourcen für den Bereich zu unterstützen. Weitere Informationen finden Sie unter [Die Rolle des Hauptbenutzers](#).
- **Eingeschränkter Zugriff** — Diese Rolle wird standardmäßig Benutzern zugewiesen, die dem Bereich beitreten, indem sie Einladungen zu Projekten im Bereich annehmen. Projektmitgliedern

wird eine Rolle in einem Projekt zugewiesen. Informationen zur Verwaltung von Projektmitgliedern finden Sie unter [Verwaltung von Projektmitgliedern](#).

In der Tabelle Space-Administratoren sind Benutzer mit der Rolle Space-Administrator aufgeführt. Diese Benutzer werden nicht in den Space-Mitgliedern angezeigt, da sie automatisch (implizit) allen Projekten im Space zugewiesen werden und keine Rolle in einem Projekt haben.

In der Tabelle Space-Mitglieder werden alle Mitglieder im Space angezeigt, die eine Rolle in einem Projekt haben, aber nicht über die Rolle des Space-Administrators verfügen.

Benutzer werden je nachdem, ob der Benutzer die Rolle des Space-Administrators innehat, CodeCatalyst wie folgt angezeigt:

- Ein Benutzer mit der Rolle Space-Administrator, der später eine Einladung und Rolle zu einem Projekt annimmt, wird nicht in der Tabelle Space-Mitglieder unter Spaces oder in der Tabelle Projektmitglieder unter Projekte angezeigt. Sie werden weiterhin an beiden Stellen in der Tabelle Space-Administratoren angezeigt. In jedem Projekt werden alle Benutzer mit der Rolle Space-Administrator in der Tabelle der Space-Administratoren des Projekts für dieses Projekt angezeigt.
- Ein Benutzer, der eine Projekteinladung zur Teilnahme mit einer Projektrolle annimmt, wird dem Bereich mit der Rolle „Eingeschränkter Zugriff“ hinzugefügt. Wenn die Rolle des Benutzers später in die Rolle des Space-Administrators wechselt, aber auch von der Tabelle Space-Mitglieder in die Tabelle Space-Administratoren wechselt. Unter dem Projekt wechselt der Benutzer von der Tabelle Projektmitglieder zur Tabelle Space-Administratoren.

Um Benutzer und Rollen in Ihrem Bereich anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.

 Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und dann Mitglieder aus.

Benutzer, die Mitglieder des Bereichs sind, werden in der Tabelle Space-Mitglieder angezeigt.

**i** Tip

Wenn Sie die Rolle eines Space-Administrators haben, können Sie sehen, zu welchen Projekten Sie direkt eingeladen wurden. Navigieren Sie zu den Projekteinstellungen für das Projekt und wählen Sie dann Meine Projekte aus.

In der Spalte Status sind die folgenden Werte gültig:

- **Eingeladen** — Die Einladung wurde CodeCatalyst gesendet, aber der Benutzer hat sie noch nicht angenommen oder abgelehnt.
- **Mitglied** — Der Benutzer hat die Einladung angenommen.

## Einen Benutzer direkt in einen Bereich einladen

Sie können Benutzer direkt in Ihren CodeCatalyst Bereich einladen. Dies ist nützlich, wenn Sie diesen Benutzer einladen möchten, Ihnen bei der Verwaltung des Bereichs zu helfen, indem Sie ihm die Rolle Space-Administrator oder Hauptbenutzer zuweisen. Wenn Sie anderen Benutzern eine dieser Rollen zuweisen, können Sie die Verantwortung für die Verwaltung des Bereichs auf mehrere Personen verteilen, ohne diese Benutzer zu Projekten einladen zu müssen.

**i** Note

Sie benötigen die Rolle eines Space-Administrators oder eines Hauptbenutzers, um Mitglieder einladen zu können.

In der Tabelle Space-Administratoren sind Benutzer mit der Rolle Space-Administrator aufgeführt. Diese Benutzer werden in der Tabelle Space-Mitglieder nicht angezeigt, da sie automatisch (implizit) allen Projekten im Space zugewiesen werden und keine Rolle in einem Projekt haben.

Mitglieder, die eine Projekteinladung annehmen, werden standardmäßig dem Bereich hinzugefügt. In der Tabelle Projektmitglieder werden alle Mitglieder des Bereichs angezeigt, die eine Rolle in einem Projekt spielen.

Weitere Informationen darüber, wie Sie eine Einladung annehmen und sich zum ersten Mal anmelden, finden Sie unter [Einrichten CodeCatalyst](#).

So laden Sie einen Benutzer in Ihren Bereich ein

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.
3. Wähle Einstellungen und dann Mitglieder aus.
4. Klicken Sie auf Einladen.
5. Gib die E-Mail-Adresse der Person ein, die du zu deinem Space einladen möchtest. Wählen Sie unter Rolle die Rolle aus, die Sie diesem Benutzer im Bereich zuweisen möchten.
6. Wählen Sie Einladen

## Eine Einladung für einen Raum stornieren

Wenn du eine Einladung zum Beitritt zu einem Space stornieren möchtest, die du kürzlich gesendet hast und die noch nicht angenommen wurde, kannst du sie stornieren.

Um Space-Einladungen verwalten zu können, benötigen Sie die Rolle Space-Administrator oder Poweruser.

Um eine Einladung eines Space-Mitglieds zu stornieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und dann Mitglieder aus.
4. Vergewissern Sie sich, dass das Mitglied den Status Eingeladen hat.

### Note

Sie können nur eine Einladung stornieren, die noch nicht angenommen wurde.

5. Wählen Sie die Option neben der Zeile mit dem eingeladenen Mitglied und wählen Sie dann Einladung stornieren aus.

6. Ein Bestätigungsfenster wird angezeigt. Wählen Sie zur Bestätigung „Einladung stornieren“.

## Die Rolle eines Space-Mitglieds ändern

Sie können die zugewiesene Rolle für ein Mitglied Ihres Bereichs ändern. Sie benötigen die Rolle eines Space-Administrators, um die Rolle eines Benutzers in dem Bereich ändern zu können.

In der Tabelle Space-Administratoren sind Benutzer mit der Rolle Space-Administrator aufgeführt. Diese Benutzer werden in der Tabelle Space-Mitglieder nicht angezeigt, da sie automatisch (implizit) allen Projekten im Space zugewiesen werden.

Um die Rolle eines Benutzers in Ihrem Bereich zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und dann Mitglieder aus.
4. Wählen Sie in der Tabelle Space-Mitglieder den Benutzer aus, dessen Rolle Sie ändern möchten. Wählen Sie Rolle ändern aus.

## Ein Space-Mitglied entfernen

Sie können ein Mitglied Ihres Spaces entfernen, wenn es nicht auf eine der Space-Ressourcen zugreifen muss. Sie benötigen die Rolle eines Space-Administrators, um ein Mitglied aus einem Space entfernen zu können.

In der Tabelle Space-Administratoren sind Benutzer mit der Rolle Space-Administrator aufgeführt. Diese Benutzer werden in der Tabelle Space-Mitglieder nicht angezeigt, da sie automatisch (implizit) allen Projekten im Space zugewiesen werden und keine Rolle in einem Projekt haben. In dieser Tabelle können Sie nur ein Mitglied Ihres Bereichs direkt entfernen.


Um einen Benutzer aus der Tabelle „Projektmitglieder“ zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich.

 Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und dann Mitglieder aus.
4. Wählen Sie den Benutzer in der Tabelle Projektmitglieder aus. Wählen Sie Remove (Entfernen) aus.


 Note

Wenn Sie ein Mitglied aus dem Bereich entfernen, wird der Benutzer aus allen Projekten im Bereich entfernt, zusammen mit den Berechtigungen, die mit den Ressourcen in diesen Projekten verknüpft sind.

## Entfernen oder Ändern der Rolle für einen Benutzer mit der Space-Administratorrolle

Sie können die Rolle eines Benutzers mit der Space-Administratorrolle für Ihren Bereich entfernen oder ändern.

Sie benötigen die Rolle eines Space-Administrators, um einen Benutzer mit der Rolle Space-Administrator aus einem Bereich entfernen zu können. Wenn Sie die Rolle für einen Benutzer mit der Rolle Space-Administrator ändern, wird der Benutzer im Wesentlichen aus der Tabelle der Space-Administratoren entfernt. Wenn dieser Benutzer in keinem Projekt im Space eine Projektkontrolle innehat, wird der Benutzer durch das Entfernen der Space-Administratorrolle aus dem Bereich entfernt.

 Note

Als Benutzer mit der Space-Administratorrolle können Sie sich nicht selbst entfernen. Wenden Sie sich an einen anderen Benutzer mit der Space-Administratorrolle.



## Um einen Benutzer mit der Rolle Space-Administrator aus der Tabelle Space-Mitglieder zu entfernen

### Note

Benutzer, die nicht explizit zu einem Projekt hinzugefügt wurden, haben keine Projektrollen (Projektadministrator oder Mitwirkender). Wenn die Rolle des Space-Administrators die einzige Rolle des Benutzers ist, wird der Benutzer vollständig aus dem Bereich entfernt.

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich, in dem Sie die Rolle eines Benutzers mit der Space-Administratorrolle entfernen oder ändern möchten.
3. Wählen Sie Einstellungen und dann Mitglieder aus.
4. Sehen Sie sich den Einladungsstatus für die Mitgliederliste an und stellen Sie sicher, dass die Liste keine unautorisierten ausstehenden Einladungen zu dem Bereich enthält (der Status „Eingeladen“).

### Important

Bevor Sie einen Benutzer mit der Space-Administratorrolle entfernen, müssen Sie sicherstellen, dass keine ausstehenden Einladungen initiiert wurden.

5. Wählen Sie den Tab Mitglieder. Wählen Sie in der Tabelle Space-Administratoren den Benutzer aus und klicken Sie dann auf Entfernen.

Führen Sie im Dialogfeld „Mitglied entfernen“ einen der folgenden Schritte aus.

- Wählen Sie die Option, um nur die Space-Administratorrolle des Benutzers zu entfernen. Wählen Sie Remove (Entfernen) aus.

### Important

Wenn dem Benutzer keine andere Rolle zugewiesen wurde, wird der Benutzer durch Ändern der Rolle von Space-Administrator aus dem Bereich entfernt.

- Wählen Sie die Option, um einen Benutzer mit der Rolle Space-Administrator aus dem Space und all seinen Projekten zu entfernen. Wählen Sie Remove (Entfernen) aus.

6. Aktualisieren Sie den Tab Mitglieder. Der Benutzer wird automatisch zur Liste der Projektmitglieder in allen Projekten hinzugefügt, in denen er über Projektrollen Mitglied war. Wenn die Rolle des Space-Administrators die einzige Rolle des Benutzers war, wird der Benutzer vollständig aus dem Bereich entfernt.

## Teams managen

Nachdem Sie einen Bereich erstellt haben, können Sie Teams hinzufügen. Teams ermöglichen es Ihnen, Benutzer so zu gruppieren, dass sie Berechtigungen gemeinsam nutzen und Projekte, Problemverfolgung, Rollen und Ressourcen verwalten können CodeCatalyst.

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

### Themen

- [Ein Team erstellen](#)
- [Ein Team ansehen](#)
- [Verwaltung der Space-Rolle für ein Team](#)
- [Eine Projektrolle für ein Team verwalten](#)
- [Direktes Hinzufügen eines Benutzers zu einem Team](#)
- [Direktes Entfernen eines Benutzers aus einem Team](#)
- [Einem Team eine SSO-Gruppe hinzufügen](#)
- [Ein Team löschen](#)

## Ein Team erstellen

Ein Team kann über Rollenberechtigungen, wie z. B. Hauptbenutzer, in einem Bereich verfügen. Ein Team kann in einem Projekt auch über Projektberechtigungen wie Projektadministrator verfügen. Teams können vielen Projekten mit unterschiedlichen Rollen für jedes Projekt zugeordnet werden. Sie können Teams verwalten, bei denen es sich bei den Teammitgliedern entweder um einzelne Benutzer für einen AWS Builder-ID-Bereich oder um SSO-Gruppen für einen Bereich handelt, der Identitätsverbund unterstützt.

Auf der Mitgliederseite für Space- und Projektbenutzer können Benutzer mehrere Rollen haben. Benutzern mit mehreren Rollen wird ein Indikator angezeigt, wenn sie mehrere Rollen haben, und sie werden mit der Rolle mit den meisten Berechtigungen zuerst angezeigt.

 Note

Wenn Ihr Bereich den Identitätsverbund unterstützt, müssen Sie Ihre SSO-Benutzer oder Ihre SSO-Gruppen bereits in IAM Identity Center eingerichtet haben.


Wie Sie Teammitglieder verwalten, hängt davon ab, wie Sie Benutzer hinzufügen und entfernen. Es gibt zwei Optionen für die Verwaltung von Teammitgliedern:

- **Direktes Hinzufügen von Benutzern** — Sie fügen Benutzer einzeln hinzu oder entfernen sie. Sie fügen beispielsweise Benutzer zu einem Team hinzu, indem Sie entweder AWS Builder ID-Benutzer oder SSO-Benutzer auswählen, die bereits in IAM Identity Center eingerichtet sind. Wenn Sie Teammitglieder verwalten möchten, indem Sie AWS Builder ID-Benutzer oder SSO-Benutzer direkt hinzufügen, ist die Option zur Verwendung von SSO-Gruppen nicht mehr verfügbar.
- **SSO-Gruppen verwenden** — Sie verwalten Teammitglieder über SSO-Gruppen, die bereits in IAM Identity Center eingerichtet sind. Wenn Sie Teammitglieder mithilfe von SSO-Gruppen verwalten möchten, ist die Option zum direkten Hinzufügen von Benutzern nicht mehr verfügbar.

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

Um ein Team zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie Team erstellen aus.
4. Geben Sie im Feld Teamname einen aussagekräftigen Namen für Ihr Team ein.

 Note


Der Teamname muss in Ihrem Bereich eindeutig sein.

(Optional) Geben Sie unter Teambeschreibung eine Beschreibung für Ihr Team ein.

5. Wählen Sie unter Space-Rolle eine Rolle aus der Liste der verfügbaren Space-Rollen aus CodeCatalyst, die Sie dem Team zuweisen möchten. Die Rolle wird von allen Mitgliedern des Teams übernommen.


- Space-Administrator — Einzelheiten finden Sie unter [Rolle „Space-Administrator“](#).
  - Eingeschränkter Zugriff — Einzelheiten finden Sie unter [Rolle mit beschränktem Zugriff](#).
  - Hauptbenutzer — Einzelheiten finden Sie unter [Die Rolle des Hauptbenutzers](#).
6. Wählen Sie unter Teammitgliedschaft eine der folgenden Optionen aus, um die Methode zum Hinzufügen von Mitgliedern zum Team auszuwählen.
- Wählen Sie Mitglieder direkt hinzufügen aus, um Benutzer einzeln zu verwalten. Dazu gehört das Hinzufügen von AWS Builder ID-Benutzern für einen Bereich oder das Hinzufügen von SSO-Benutzern für einen Bereich, der Identitätsverbund unterstützt.
  - Wählen Sie SSO-Gruppen verwenden, um SSO-Gruppen auszuwählen, die Sie bereits in IAM Identity Center eingerichtet haben.

Wählen Sie unter SSO-Gruppen das Kästchen neben den Gruppen aus, die Sie hinzufügen möchten. Sie können bis zu fünf SSO-Gruppen hinzufügen.

 Note

Sie können dies später nicht ändern. Wenn Sie Teammitglieder verwalten möchten, indem Sie AWS Builder ID-Benutzer oder SSO-Benutzer direkt hinzufügen, ist die Option zur Verwendung von SSO-Gruppen nicht mehr verfügbar. Wenn Sie Teammitglieder mithilfe von SSO-Gruppen verwalten möchten, ist die Option zum direkten Hinzufügen von Benutzern nicht mehr verfügbar.

7. Wählen Sie Erstellen.

 Note

Wenn Sie sich für die Verwendung von SSO-Gruppen entscheiden, beachten Sie, dass die Benutzer in der SSO-Gruppe nicht bei der Erstellung des Teams abgerufen werden. Die Benutzer müssen sich angemeldet haben, CodeCatalyst bevor sie in der Liste angezeigt werden.

## Ein Team ansehen

CodeCatalystIn können Sie die Projekte und Rollen Ihres Teams einsehen. Auf der Mitgliederseite können Sie die Projektrollen und eine Liste der Benutzer einsehen. Für Teams vom Typ SSO-Gruppe wird Ihnen auch eine Liste der SSO-Gruppen angezeigt, die dem Team zugeordnet sind.

Um ein Team anzusehen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Sehen Sie sich unter Space-Rolle die Rolle an, die dem Team für diesen Bereich zugewiesen wurde.
4. Sehen Sie sich auf der Registerkarte Projektrollen das Projekt und die Projektrolle an, die dem Team für jedes CodeCatalyst Projekt in dem Bereich zugewiesen wurden, in dem das Team als Mitglied hinzugefügt wurde (nur für einen AWS Builder-ID-Bereich).
5. Sehen Sie sich auf der Registerkarte Mitglieder die Liste der Mitglieder an, die dem Team zugewiesen sind.
6. Sehen Sie sich auf der Registerkarte SSO-Gruppen die Liste der SSO-Gruppen an, die dem Team zugewiesen sind (für einen Bereich, der nur Identitätsverbund unterstützt).

## Verwaltung der Space-Rolle für ein Team

Ein Team kann über Rollenberechtigungen, wie z. B. Hauptbenutzer, in einem Bereich verfügen. Sie können die Space-Rolle für ein Team ändern, beachten Sie jedoch, dass alle Mitglieder des Teams diese Berechtigungen erben.

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

Die Space-Rolle für ein Team ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie unter Aktionen die Option Bereichsrolle ändern aus. Sie können die Space-Rolle in eine der folgenden Rollen ändern. Dadurch wird die Rolle für alle Mitglieder des Teams geändert.
  - Space-Administrator — Einzelheiten finden Sie unter [Rolle „Space-Administrator“](#).

- Eingeschränkter Zugriff — Einzelheiten finden Sie unter [Rolle mit beschränktem Zugriff](#).
  - Hauptbenutzer — Einzelheiten finden Sie unter [Die Rolle des Hauptbenutzers](#).
4. Wählen Sie Speichern.

## Eine Projektrolle für ein Team verwalten

Ein Team in CodeCatalyst ähnelt einem Benutzer insofern, als die Teammitglieder in einem Projekt über Rollenberechtigungen wie Projektadministrator verfügen können. Eine Rollenänderung wird auf das Team angewendet, und alle Teammitglieder erben diese Berechtigungen. Sie können für jedes Projekt eine Rolle auswählen, die dem Team automatisch zugewiesen wird.

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

Um eine Projektrolle hinzuzufügen oder zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie den Tab Projektrollen aus.
4. Um eine Rolle zu ändern, wählen Sie in dieser Liste den Selektor neben dem Projekt aus und wählen Sie dann Rolle ändern aus. Um eine Rolle hinzuzufügen, wählen Sie Projektrolle hinzufügen aus. Wählen Sie unter Projekt das Projekt aus, das Sie hinzufügen möchten, und wählen Sie unter Rolle die Rolle aus. Wählen Sie eine der verfügbaren Projektrollen aus:
  - Projektadministrator — Einzelheiten finden Sie unter [Rolle als Projektadministrator](#).
  - Mitwirkender — Einzelheiten finden Sie unter [Rolle „Mitwirkender“](#).
  - Prüfer — Einzelheiten finden Sie unter [Rolle des Prüfers](#).
  - Schreibgeschützt — Einzelheiten finden Sie unter [Rolle „Nur lesen“](#).
5. Wählen Sie Speichern.

Um eine Projektrolle zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie den Tab Projektrollen aus.
4. Wählen Sie die Rolle aus, die Sie entfernen möchten.

**⚠ Important**

Wenn Sie eine Rolle aus einem Team entfernen, werden die zugehörigen Berechtigungen für alle Benutzer im Team entfernt.

5. Wählen Sie Speichern.

## Direktes Hinzufügen eines Benutzers zu einem Team

Sie können Ihrem Team Teammitglieder hinzufügen. Wenn Sie einen Benutzer hinzufügen, erbt der neue Benutzer die Berechtigungen aller vorhandenen Rollen im Team.

Unabhängig davon, ob Ihr Bereich für AWS Builder ID-Benutzerunterstützung oder Identitätsverbund eingerichtet ist, können Sie Ihren Bereich so einrichten, dass Benutzer direkt hinzugefügt werden.

**ℹ Note**

Wenn Ihr Bereich für die Verwaltung von Teammitgliedern mithilfe von SSO-Gruppen eingerichtet ist, ist die Option „Benutzer direkt hinzufügen“ nicht verfügbar. Informationen zur Verwendung von SSO-Gruppen finden Sie unter [Einem Team eine SSO-Gruppe hinzufügen](#).

Sie müssen über die Rolle eines Space-Administrators verfügen, um Teams verwalten zu können.

Um einen Benutzer direkt hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie den Tab Mitglieder.
4. Wählen Sie Mitglied hinzufügen.

**ℹ Note**

Benutzer, die zu einem Team hinzugefügt werden, müssen bereits Mitglieder eines Bereichs sein. Sie können kein Teammitglied hinzufügen oder einladen, das kein Mitglied des Bereichs ist.

5. Wählen Sie im Dropdown-Feld einen Benutzer aus und klicken Sie dann auf Speichern. Wählen Sie entweder AWS Builder ID-Benutzer oder SSO-Benutzer aus, die bereits in IAM Identity Center eingerichtet sind.

## Direktes Entfernen eines Benutzers aus einem Team

Sie können Teammitglieder aus Ihrem Team entfernen. Alle Berechtigungen werden nicht mehr vom Benutzer vererbt. Sie können den Benutzer später wieder zum Team hinzufügen.

### Note

Wenn Sie ein Teammitglied entfernen, werden die zugehörigen Berechtigungen für den Benutzer aus allen Projekten und Ressourcen im Bereich entfernt.

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

So entfernen Sie ein Teammitglied

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie den Tab Mitglieder.
4. Wählen Sie den Selektor neben dem Benutzer aus, den Sie entfernen möchten, und wählen Sie dann Entfernen aus.
5. Geben Sie Entfernen in das Eingabefeld ein und wählen Sie dann Entfernen aus.

## Einem Team eine SSO-Gruppe hinzufügen

Wenn Ihr Bereich als Bereich mit SSO-Benutzern und -Gruppen konfiguriert ist, die im IAM Identity Center verwaltet werden, können Sie eine SSO-Gruppe hinzufügen, die dem Bereich als separates Team beitrifft.

### Note

Wenn Sie Teammitglieder verwalten möchten, indem Sie AWS Builder ID-Benutzer oder SSO-Benutzer direkt hinzufügen, ist die Option zur Verwendung von SSO-Gruppen nicht



verfügbar. Informationen zum direkten Hinzufügen von Benutzern finden Sie unter [Direktes Hinzufügen eines Benutzers zu einem Team](#).

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

Um eine SSO-Gruppe als Team hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wähle auf der Seite für deinen Bereich Teams aus. Wählen Sie den Tab SSO-Gruppen aus.
3. Wählen Sie die SSO-Gruppen aus, die Sie hinzufügen möchten. Sie können bis zu fünf SSO-Gruppen hinzufügen.

## Ein Team löschen

Sie können ein Team löschen, das Sie nicht mehr benötigen.

### Note

Wenn Sie ein Team löschen, werden die zugehörigen Berechtigungen für alle Teammitglieder aus allen Projekten und Ressourcen im Bereich entfernt.

Sie benötigen die Rolle eines Space-Administrators, um Teams verwalten zu können.

Lösche ein Team

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wähle Einstellungen und dann Teams aus.
3. Wählen Sie unter Aktionen die Option Team löschen aus. Dadurch ändert sich die Rolle für das gesamte Team.
4. Wählen Sie Löschen.

## Verwaltung von Maschinenressourcen

Eine Maschinenressource stellt Ihre Identität gegenüber Ihrer autorisierten Ressource dar, wenn Sie CodeCatalyst über SSO darauf zugreifen. Maschinenressourcen werden verwendet, um

Berechtigungen für Ressourcen im Bereich zu erteilen, z. B. Blueprints und Workflows. Sie können die Maschinenressourcen in Ihrem Bereich anzeigen und wählen, ob Sie Maschinenressourcen für Ihren Bereich aktivieren oder deaktivieren möchten. Möglicherweise möchten Sie eine Maschinenressource deaktivieren, um den Zugriff zu verwalten, und sie dann später wieder aktivieren.

Diese Operationen sind für Maschinenressourcen verfügbar, wenn eine Maschinenressource gesperrt oder deaktiviert werden muss. Wenn Sie beispielsweise vermuten, dass Anmeldeinformationen kompromittiert wurden, können Sie die Computerressource deaktivieren. Im Allgemeinen müssen diese Operationen nicht verwendet werden.

Sie benötigen die Rolle eines Space-Administrators, um diese Seite anzeigen und Maschinenressourcen auf Space-Ebene verwalten zu können.

Themen

- [Maschinenressourcen anzeigen](#)
- [Deaktivieren Sie die Maschinenressourcen](#)
- [Aktivieren Sie Maschinenressourcen](#)

## Maschinenressourcen anzeigen

Sie können eine Liste der Maschinenressourcen anzeigen, die in Ihrem Bereich verwendet werden.

Sie benötigen die Rolle eines Space-Administrators, um Maschinenressourcen verwalten zu können.

Um Maschinenressourcen anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich und wähle dann Einstellungen. Wählen Sie Maschinenressourcen aus.
3. Wählen Sie in der Dropdownliste Workflow-Aktion aus, um nur die Maschinenressourcen für Workflows anzuzeigen. Wählen Sie Blueprint aus, um nur die Maschinenressourcen für Blueprints anzuzeigen.

Sie können mithilfe des Felds Filter auch nach einem Namen filtern.

## Deaktivieren Sie die Maschinenressourcen

Sie können wählen, ob Sie Maschinenressourcen deaktivieren möchten, die in Ihrem Bereich verwendet werden.

### Important

Durch das Deaktivieren von Maschinenressourcen werden alle Berechtigungen für alle zugehörigen Blueprints oder Workflows im Bereich entfernt.

Sie benötigen die Rolle eines Space-Administrators, um Maschinenressourcen verwalten zu können.

Um Maschinenressourcen zu deaktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich und wähle dann Einstellungen. Wählen Sie Maschinenressourcen aus.
3. Wählen Sie eine der folgenden Optionen aus.

### Important

Durch das Deaktivieren von Maschinenressourcen werden alle Berechtigungen für alle zugehörigen Blueprints oder Workflows im Bereich entfernt.

- Um sie einzeln zu deaktivieren, wählen Sie den Selektor neben einer oder mehreren Maschinenressourcen aus, die Sie deaktivieren möchten. Wählen Sie Deaktivieren und anschließend Diese Ressource aus.
- Um alle Ressourcen zu deaktivieren, wählen Sie „Deaktivieren“ und anschließend „Alle Ressourcen“.
- Um alle Workflow-Aktionen zu deaktivieren, wählen Sie Deaktivieren und dann Alle Workflow-Aktionen aus.
- Um alle Blueprints zu deaktivieren, wählen Sie Deaktivieren und dann Alle Blueprints aus.

## Aktivieren Sie Maschinenressourcen

Sie können wählen, ob Maschinenressourcen aktiviert werden sollen, die in Ihrem Bereich verwendet werden und die deaktiviert wurden.

Sie benötigen die Rolle eines Space-Administrators, um Maschinenressourcen verwalten zu können.

Um Maschinenressourcen zu aktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich und wähle dann Einstellungen. Wählen Sie Maschinenressourcen aus.
3. Wählen Sie eine der folgenden Optionen aus.
  - Um sie einzeln zu aktivieren, wählen Sie den Selektor neben einer oder mehreren Maschinenressourcen, die Sie aktivieren möchten. Wählen Sie Aktivieren und anschließend Diese Ressource aus.
  - Um alle Ressourcen zu aktivieren, wählen Sie Aktivieren und dann Alle Ressourcen aus.
  - Um alle Workflow-Aktionen zu aktivieren, wählen Sie Aktivieren und dann Alle Workflow-Aktionen aus.
  - Um alle Blueprints zu aktivieren, wählen Sie Aktivieren und dann Alle Blueprints aus.

## Verwaltung von Entwicklungsumgebungen für einen Bereich

Alle Entwicklungsumgebungen werden als Teil eines Projekts in einem Bereich erstellt. Space-Mitglieder können ihre eigenen Entwicklungsumgebungen innerhalb eines Projekts auf der Ebene des Quell-Repositorys erstellen. Space-Administratoren können dann die CodeCatalyst Amazon-Konsole verwenden, um Entwicklungsumgebungen im Namen von Space-Mitgliedern anzusehen, zu bearbeiten, zu löschen und zu beenden. Kurz gesagt, Space-Administratoren verwalten Entwicklungsumgebungen auf Bereichsebene.

Überlegungen zur Verwaltung von Entwicklungsumgebungen

- Sie benötigen die Rolle eines Space-Administrators, um die Seite Entwicklungsumgebungen unter Einstellungen aufrufen und Entwicklungsumgebungen auf Space-Ebene verwalten zu können.

- Space-Mitglieder verwalten die Entwicklungsumgebungen, die sie in Projekten erstellen, über ihre CodeCatalyst Konten. Wenn Sie Entwicklungsumgebungen als Space-Administrator verwalten, verwalten Sie diese Ressourcen im Namen der Space-Mitglieder.
- Entwicklungsumgebungen verwenden standardmäßig eine bestimmte Rechen- und Speicherkonfiguration. Informationen zur Abrechnung und zu den Tarifen für das Upgrade Ihrer Konfiguration finden Sie auf der [CodeCatalyst Amazon-Preisseite](#).

Weitere Überlegungen zu Entwicklungsumgebungen, einschließlich des Stoppens laufender Instances, der Standard-Rechenkonfiguration, der Aktualisierung Ihrer Rechenleistung, der Entstehung von Kosten und der Konfiguration von Timeouts, finden Sie unter.

[Entwicklungsumgebungen in CodeCatalyst](#)

### Themen

- [Entwicklungsumgebungen für deinen Bereich anzeigen](#)
- [Eine Entwicklungsumgebung für deinen Bereich bearbeiten](#)
- [Eine Entwicklungsumgebung für deinen Bereich beenden](#)
- [Löschen einer Entwicklungsumgebung für deinen Bereich](#)

## Entwicklungsumgebungen für deinen Bereich anzeigen

Sie können den Typ, den Status und die Details aller Entwicklungsumgebungen in Ihrem Bereich einsehen. Weitere Informationen zum Erstellen und Ausführen von Entwicklungsumgebungen finden Sie unter [Erstellen einer Entwicklungsumgebung](#).

Sie benötigen die Rolle eines Space-Administrators, um diese Seite aufrufen und Entwicklungsumgebungen auf Space-Ebene verwalten zu können.

Um Entwicklungsumgebungen in Ihrem Bereich anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und anschließend Entwicklungsumgebungen aus.

Die Seite listet alle Entwicklungsumgebungen in Ihrem Bereich auf. Sie können den Ressourcennamen, gegebenenfalls den Ressourcenalias, den IDE-Typ, die standardmäßige oder konfigurierte Compute and Storage und das konfigurierte Timeout für jede Entwicklungsumgebung einsehen.

## Eine Entwicklungsumgebung für deinen Bereich bearbeiten

Sie können die Konfiguration für eine Entwicklungsumgebung bearbeiten, z. B. die konfigurierte Dauer des Timeouts, falls vorhanden, bis eine Entwicklungsumgebung im Leerlauf nicht mehr läuft. Weitere Informationen zum Bearbeiten einer Entwicklungsumgebung finden Sie unter [Eine Entwicklungsumgebung bearbeiten](#).

Sie benötigen die Rolle eines Space-Administrators, um diese Seite aufrufen und Entwicklungsumgebungen auf Space-Ebene verwalten zu können.

Um Entwicklungsumgebungen in Ihrem Bereich zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und anschließend Entwicklungsumgebungen aus.
4. Wählen Sie den Selektor neben der Entwicklungsumgebung aus, die Sie verwalten möchten. Wählen Sie Bearbeiten aus.
5. Nehmen Sie die gewünschten Änderungen am Berechnungs- oder Inaktivitäts-Timeout für die Entwicklungsumgebung vor.
6. Klicken Sie auf Speichern.

## Eine Entwicklungsumgebung für deinen Bereich beenden

Sie können eine laufende Entwicklungsumgebung beenden, bevor sie inaktiv wird, wenn für die Entwicklungsumgebung ein Timeout konfiguriert ist. Andernfalls wird eine Entwicklungsumgebung mit einem abgelaufenen Timeout bereits gestoppt. Weitere Hinweise zum Stoppen einer Entwicklungsumgebung finden Sie unter [Anhalten einer Entwicklungsumgebung](#)

Sie benötigen die Rolle eines Space-Administrators, um diese Seite aufrufen und Entwicklungsumgebungen auf Space-Ebene verwalten zu können.

Um eine Entwicklungsumgebung in deinem Bereich zu stoppen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und anschließend Entwicklungsumgebungen aus.
4. Wählen Sie den Selektor neben der Entwicklungsumgebung, die Sie verwalten möchten. Wählen Sie Beenden aus.

## Löschen einer Entwicklungsumgebung für deinen Bereich

Sie können eine Entwicklungsumgebung löschen, die nicht mehr benötigt wird oder die keinen Besitzer mehr hat. Weitere Informationen zu Überlegungen beim Löschen einer Entwicklungsumgebung finden Sie unter [Löschen einer Entwicklungsumgebung](#).

Sie benötigen die Rolle eines Space-Administrators, um diese Seite aufrufen und Entwicklungsumgebungen auf Space-Ebene verwalten zu können.

Um Entwicklungsumgebungen in Ihrem Bereich zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

**Tip**

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie Einstellungen und anschließend Entwicklungsumgebungen aus.
4. Wählen Sie den Selektor neben der Entwicklungsumgebung aus, die Sie verwalten möchten. Wählen Sie Löschen aus. Geben `delete` Sie zur Bestätigung den Text ein und wählen Sie dann Löschen aus.

## Kontingente für Leerzeichen in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Beschränkungen für Speicherplätze in Amazon beschriebenen CodeCatalyst. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

Anzahl der aktiven Spaces pro Benutzer pro AWS-Region	Five
Anzahl der Space-Kreationen pro Region pro Monat und Nutzer	Five
Beschreibungen der Räume	<p>Raumbeschreibungen sind optional. Falls angegeben, müssen sie zwischen 0 und 200 Zeichen lang sein. Sie können eine beliebige Kombination aus Buchstaben, Zahlen, Leerzeichen, Punkten, Unterstrichen, Kommas, Bindestrichen und den folgenden Sonderzeichen enthalten:</p> <p>? &amp; \$ % + = / \ ; : \n \t \r</p>
Namen von Leerzeichen	<p>Raumnamen müssen überall eindeutig sein CodeCatalyst. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden.</p>



Raumnamen müssen zwischen 3 und 63 Zeichen lang sein. Sie müssen außerdem mit einem alphanumerischen Zeichen beginnen. Raumnamen können eine beliebige Kombination aus Buchstaben, Zahlen, Punkten, Unterstrichen und Bindestrichen enthalten. Sie dürfen keines der folgenden Zeichen enthalten:

! ? @ # \$ % ^ & \* ( ) + = { } [ ]  
| \ > < ~ ` ' " ; :

# Projekte in CodeCatalyst

Sie verwenden Projekte in Amazon CodeCatalyst, um einen Bereich für die Zusammenarbeit einzurichten, in dem Entwicklungsteams Entwicklungsaufgaben mit gemeinsamen Workflows und Repositories für Continuous Integration/Continuous Delivery (CI/CD) durchführen können. Wenn Sie ein Projekt erstellen, können Sie Ressourcen hinzufügen, aktualisieren oder entfernen. Sie können auch den Fortschritt der Arbeit Ihres Teams überwachen. Sie können mehrere Projekte in einem Bereich haben.

Leerzeichen in CodeCatalyst bestehen aus Projekten. Sie können jedes Projekt in Ihrem Bereich sehen, aber Sie können nur die Projekte verwenden, bei denen Sie Mitglied sind. Wenn Sie ein Projekt erstellen, werden Standardrollen für Ihr Projekt generiert, die Sie Benutzern zuweisen, die Sie zu Ihrem Projekt einladen.

- Jeder, dem Projekt mit einer Projektrolle, z. B. der Rolle „Mitwirkender“, zugewiesen ist, kann auf Projektressourcen zugreifen, z. B. auf ein Quell-Repository.
- Jeder Benutzer mit der Rolle „Space-Administrator“, „Projektadministrator“ oder „Projektadministrator“ kann Einladungen zur Teilnahme an einem Projekt versenden.
- Benutzer mit der Rolle „Projektadministrator“ können Aktivitäten, Status und andere Einstellungen für gemeinsam genutzte Ressourcen verfolgen.
- Benutzer mit der Rolle „Eingeschränkter Zugriff“ können Projektzuweisungen für Funktionen, Codekorrekturen und Tests im Rahmen von CI/CD-Workflows verwalten.

Workflows werden verwendet, um Anwendungen als CI/CD-Pipeline zu erstellen, zu testen und zu veröffentlichen oder zu aktualisieren. Sie können Workflows zusammenstellen, indem Sie Aktionen hinzufügen, die Ihre Quellartefakte übertragen und bearbeiten. Wenn Sie Aktionen ausführen, werden Ihre Projekt-Cloud-Ressourcen verwendet, um On-Demand-Rechenfunktionen für Ihre Workflow-Aktionen bereitzustellen. Je nach Aktivität und Ausgabe, die Sie einrichten möchten, können Sie weitere CI/CD-Workflows konfigurieren. Sie könnten beispielsweise einen Workflow nur für Build- und Testaktionen erstellen, in dem Sie Testergebnisse anzeigen und den Workflow ohne Bereitstellung abschließen können, während Sie Fehler beheben. Anschließend können Sie einen weiteren Workflow erstellen, um Ihre Anwendung zu erstellen und in einer Staging-Umgebung bereitzustellen.

Wenn Sie ein Projekt erstellen, können Sie einen Blueprint verwenden, um ein Projekt zu erstellen, das Beispielcode enthält und Ressourcen erstellt, oder Sie können mit einem leeren Projekt

beginnen. Wenn Sie ein Projekt mithilfe eines Blueprints erstellen, bestimmt der von Ihnen gewählte Blueprint, welche Ressourcen Ihrem Projekt hinzugefügt werden und welche Tools CodeCatalyst erstellt oder konfiguriert werden, damit Sie Ihre Projektressourcen verfolgen und verwenden können. Sie können Ressourcen manuell hinzufügen oder entfernen, nachdem Sie ein Projekt erstellt haben. Die folgenden Ressourcen können erstellt oder konfiguriert werden von CodeCatalyst:

- Probleme — Sie verfolgen die Arbeit im Zusammenhang mit Ihrem Projekt in einem separaten Datensatz, den sogenannten Problemen. Sie können Probleme für Funktionen, Aufgaben, Fehler und jede andere Arbeit für Ihre Projekte angeben.
- Benachrichtigungen — Sie konfigurieren Benachrichtigungen, indem Sie die Ressource auswählen, die Sie verfolgen möchten, das Ereignis, nach dem Sie Ausschau halten möchten, und den Zielclient oder die Ziel-E-Mail, über die Sie Benachrichtigungen erhalten möchten.
- Suche — Du kannst dein Projekt nach Code, Issues, Benutzern, Pull Requests und Paketen durchsuchen. Du kannst in einem einzelnen Projekt oder in allen Projekten suchen.
- Quell-Repository — Sie arbeiten mit Ihrem Quellcode in einem Repository für Ihr Projekt. Wenn du eine Quellcode-Änderung festlegst oder einen Pull-Request in einem bestimmten Branch zusammenführst, wird dein Quellcode CodeCatalyst aktualisiert.

Jedes Projekt verfolgt die Projektaktivitäten in Form einer Liste von Ereignissen nach Benutzern, z. B. wenn ein Projekt erstellt oder eine Ressource geändert wird. Die Projektaktivität wird auf Bereichsebene überwacht und aggregiert. Weitere Informationen zum Arbeiten mit Aktivitätsdaten finden Sie unter [Alle Projekte ansehen](#).

Wenn Ihr Projekt AWS Ressourcen verwendet, können Sie Ihr CodeCatalyst Konto mit einem AWS Konto verbinden, für das Sie über Administratorrechte verfügen, um Ressourcen für Ihr Projekt zu integrieren.

Sie können Ihrem Projekt Quell-Repositorys, Probleme und andere Ressourcen hinzufügen, nachdem Sie es erstellt haben. Sie benötigen die Rolle eines Space-Administrators, um Projekte erstellen zu können.

## Ein Projekt in Amazon erstellen CodeCatalyst

Mit CodeCatalyst Projekten können Sie Entwicklungsaufgaben mit gemeinsamen Workflows und Repositorys für kontinuierliche Integration/Continuous Delivery (CI/CD) ausführen, Ressourcen verwalten, Probleme verfolgen und Benutzer hinzufügen.

Bevor Sie ein Projekt erstellen können, müssen Sie über die Rolle eines Space-Administrators oder eines Hauptbenutzers verfügen.

## Themen

- [Ein Projekt mit einem Blueprint erstellen](#)
- [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#)
- [Ein Projekt mit einem verknüpften GitHub Repository erstellen](#)

## Ein Projekt mit einem Blueprint erstellen

Sie können all Ihre Projektressourcen und den Beispielcode mit einem Projekt-Blueprint bereitstellen. Informationen zu Blueprints finden Sie in der Blueprint-Referenz unter. [Referenz zum Projekt-Blueprint](#)

Um ein Projekt mit einem Blueprint zu erstellen

1. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
2. Wählen Sie im Space-Dashboard die Option Projekt erstellen aus.
3. Wählen Sie Mit einem Blueprint beginnen aus.
4. Wählen Sie einen Blueprint und dann Weiter aus.
5. Geben Sie unter Geben Sie Ihrem Projekt einen Namen den Namen ein, den Sie Ihrem Projekt zuweisen möchten, sowie die zugehörigen Ressourcennamen. Der Name muss in Ihrem Bereich eindeutig sein.
6. Konfigurieren Sie unter Projektressourcen allgemeine Projektparameter.
7. (Optional) Um Definitionsdateien mit Aktualisierungen auf der Grundlage der von Ihnen ausgewählten Projektparameter anzuzeigen, wählen Sie unter Projektvorschau generieren die Option Code anzeigen oder Workflow anzeigen aus.
8. (Optional) Wählen Sie auf der Karte des Blueprints die Option Details anzeigen aus, um spezifische Details zum Blueprint anzuzeigen, z. B. einen Überblick über die Architektur des Blueprints, die erforderlichen Verbindungen und Berechtigungen sowie die Art der Ressourcen, die der Blueprint erstellt.
9. Wählen Sie Create project (Projekt erstellen) aus.

Weitere Informationen zu Projekt-Blueprints finden Sie unter. [Referenz zum Projekt-Blueprint](#)

## Ein leeres Projekt in Amazon erstellen CodeCatalyst

Sie können ein leeres Projekt ohne Ressourcen erstellen und die gewünschten Ressourcen zu einem späteren Zeitpunkt manuell hinzufügen.

Bevor Sie ein Projekt erstellen können, müssen Sie über die Rolle Space-Administrator oder Poweruser verfügen.

Um ein leeres Projekt zu erstellen

1. Navigieren Sie zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
2. Wählen Sie im Space-Dashboard die Option Projekt erstellen aus.
3. Wählen Sie „Von vorne beginnen“.
4. Geben Sie unter Geben Sie Ihrem Projekt einen Namen ein den Namen ein, den Sie Ihrem Projekt zuweisen möchten. Der Name muss in Ihrem Bereich eindeutig sein.
5. Wählen Sie Create project (Projekt erstellen) aus.

## Ein Projekt mit einem verknüpften GitHub Repository erstellen

Sie können ein neues CodeCatalyst Projekt erstellen, das auf ein GitHub Quell-Repository verweist. Sie können dann Ihr verknüpftes GitHub Quell-Repository in Ihrem CodeCatalyst Projekt verwenden.

Bevor Sie ein CodeCatalyst Projekt erstellen können, müssen Sie über die Rolle Space-Administrator oder Poweruser verfügen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#) und [Einen Benutzer direkt in einen Bereich einladen](#).

Sie müssen bereits über ein GitHub Konto verfügen und das Repository, zu dem Sie eine Verknüpfung herstellen möchten, bereits erstellt haben.

Um ein Projekt zu erstellen CodeCatalyst , das auf ein Quell-Repository in Ihrem GitHub Konto verweist, müssen Sie die folgenden drei Aufgaben ausführen:

1. Installieren Sie die GitHub Repository-Erweiterung.
2. Connect dein GitHub Konto mit CodeCatalyst.
3. Erstellen Sie ein CodeCatalyst Projekt, das mit Ihrem GitHub Konto verknüpft ist.

## Um die GitHub Repository-Erweiterung zu installieren

1. Navigieren Sie zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
2. Wählen Sie im Space-Dashboard die Option Projekt erstellen aus.
3. Wählen Sie Bring Your Own Code aus.

Wenn die GitHub Repository-Erweiterung noch nicht installiert ist, wird eine Installationsaufforderung angezeigt.

4. Wählen Sie Installieren aus. Überprüfen Sie die für die Erweiterung erforderlichen Berechtigungen. Wenn Sie fortfahren möchten, wählen Sie erneut Installieren aus.

Nachdem Sie die GitHub Repository-Erweiterung installiert haben, besteht der nächste Schritt darin, Ihr GitHub Konto mit Ihrem CodeCatalyst Bereich zu verbinden.

### Note

Du kannst leere oder archivierte GitHub Repositories nicht für Projekte verwenden. CodeCatalyst Die GitHub Repository-Erweiterung ist nicht mit GitHub Enterprise Server-Repositories kompatibel.

## Um Ihr GitHub Konto zu verbinden CodeCatalyst

1. Wenn auf der Seite Projekt erstellen in CodeCatalyst kein GitHub Konto verbunden ist, wird eine Aufforderung angezeigt. Wählen Sie GitHub Konto Connect, um zur externen Site zu wechseln GitHub.
2. Melden Sie sich mit Ihren GitHub Anmeldeinformationen bei Ihrem GitHub Konto an und wählen Sie dann das Konto aus, auf dem Sie Amazon installieren möchten CodeCatalyst.

### Tip

Wenn Sie zuvor ein GitHub Konto mit dem Bereich verbunden haben, werden Sie nicht aufgefordert, es erneut zu autorisieren. Stattdessen wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, wo Sie die Erweiterung installieren möchten, wenn Sie Mitglied oder Mitarbeiter in mehr als einem GitHub Bereich sind, oder die Konfigurationsseite für die CodeCatalyst Amazon-Anwendung, wenn Sie nur zu einem GitHub Bereich gehören. Konfigurieren Sie die Anwendung für den Repository-Zugriff, den Sie zulassen

möchten, und wählen Sie dann Speichern. Wenn die Schaltfläche Speichern nicht aktiv ist, nehmen Sie eine Änderung an der Konfiguration vor, und versuchen Sie es erneut.

3. Wählen Sie aus, ob Sie den CodeCatalyst Zugriff auf alle aktuellen und future Repositorys zulassen möchten, oder wählen Sie die spezifischen GitHub Repositorys aus, in denen Sie sie verwenden möchten. CodeCatalyst Die Standardoption besteht darin, alle GitHub Repositorys in das GitHub Konto aufzunehmen, einschließlich future Repositorys, auf die zugegriffen wird. CodeCatalyst
4. Überprüfen Sie die Berechtigungen, die Sie erhalten haben CodeCatalyst, und wählen Sie dann Installieren.

Nachdem Sie Ihr GitHub Konto mit verbunden haben CodeCatalyst, können Sie die GitHub Repositorys dieses Kontos mit Ihren CodeCatalyst Projekten verknüpfen.

So erstellen Sie Ihr Projekt

1. Führen Sie auf der Seite Projekt erstellen im Dropdownmenü GitHubKonto einen der folgenden Schritte aus:
  - Wählen Sie ein GitHub Konto aus, mit dem Sie bereits eine Verbindung hergestellt haben. CodeCatalyst
  - (Optional) Wenn Sie das GitHub Konto, das Sie verwenden möchten, nicht sehen, wählen Sie GitHub Konto Connect aus, um zu der Seite für Erweiterungen zu gelangen CodeCatalyst. Weitere Informationen finden Sie unter [Verwendung von GitHub Repositorys in CodeCatalyst](#).
2. Im GitHub Repository-Dropdownmenü werden die GitHub Repositorys für Ihr verbundenes GitHub Konto in der Dropdownliste angezeigt. Wählen Sie das GitHub Repository aus, das Sie mit Ihrem Projekt verknüpfen möchten.
3. Geben Sie im Texteingabefeld Name Ihres Projekts den Namen ein, den Sie Ihrem Projekt zuweisen möchten. Der Name muss in Ihrem Bereich eindeutig sein.
4. Wählen Sie Create project (Projekt erstellen) aus.

Sobald Ihr Projekt fertig ist, können Sie Ressourcen und Aufgaben hinzufügen.

- Weitere Informationen zu den CI/CD-Workflows, die mit Ihrem Projekt erstellt wurden, finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#)

- Informationen zum Arbeiten mit Build-Aktionen, die denen in Ihrem neuen Projekt ähneln und Build-Artefakte in einem Amazon S3 S3-Bucket bereitstellen, finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#) und [Tutorial: Artefakte auf Amazon S3 hochladen](#).
- Informationen dazu, wie Sie mit einem leeren Projekt beginnen und mit der Bereitstellung einer ähnlichen serverlosen Anwendung mit einer AWS CloudFormation Stack-Bereitstellung arbeiten, finden Sie unter [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#).
- Informationen zum Hinzufügen einer Planungstafel für Probleme finden Sie unter [Probleme in CodeCatalyst](#).
- Informationen zur Projektübersicht, zum Projektstatus, zu den letzten Teamaktivitäten und zur zugewiesenen Arbeit finden Sie unter [Ein Projekt ansehen](#).
- Informationen zum Anzeigen des Quellcodes oder zum Erstellen eines Pull-Requests finden Sie unter [Quell-Repositoryn in CodeCatalyst](#).
- Informationen zum Einrichten von Benachrichtigungen, die Statuswarnungen bei erfolgreicher oder fehlgeschlagener Workflow-Ausführung senden, finden Sie unter [Benachrichtigungen in Amazon verwalten CodeCatalyst](#).
- Informationen zum Einladen von Mitgliedern zu Ihrem Projekt finden Sie unter [Verwaltung von Projektmitgliedern](#).
- Informationen zum Einrichten von Entwicklungsumgebungen finden Sie unter [Entwicklungsumgebungen in CodeCatalyst](#).

## Ein Projekt ansehen

In Ihrem CodeCatalyst Bereich können Sie Details zu jedem Projekt einsehen, für das Sie über Projektberechtigungen verfügen.

Um ein Projekt ansehen zu können, müssen Sie Mitglied des Projekts sein oder die Rolle eines Space-Administrators für den Bereich besitzen.

Wenn Sie noch kein Projekt erstellt haben, finden Sie weitere Informationen unter [Ein Projekt in Amazon erstellen CodeCatalyst](#). Sie müssen über die Rolle eines Space-Administrators für den Bereich verfügen, in dem Sie ein Projekt erstellen möchten.

- In der Projektübersicht können Sie Projektmitglieder, Quell-Repositorys, Workflow-Ausführungen, offene Pull-Requests, Projektentwicklungsumgebungen und Probleme einsehen.



- Unter Projekteinstellungen kannst du Projektdetails anzeigen und verwalten, das Projekt löschen, neue Mitglieder zum Projekt einladen, Projektmitglieder verwalten und Benachrichtigungen konfigurieren.

## Projektaufgaben und Entwicklungsumgebungen anzeigen

Verwenden Sie die Konsole, um eine Zusammenfassung der Projektaufgaben, wie z. B. offene Issues und Pull Requests, die Ihnen zugewiesen oder von Ihnen erstellt wurden, sowie der zugehörigen Entwicklungsumgebungen des Projekts anzuzeigen.

Um ein Projekt anzeigen zu können, müssen Sie Mitglied des Projekts sein oder die Rolle eines Space-Administrators für den Bereich besitzen.

Um deine Quell-Repositorys, Workflow-Ausführungen, Issues, Pull Requests, Entwicklungsumgebungen und Issues anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit dem Projekt, das Sie anzeigen möchten. Wählen Sie unter Projekte Ihr Projekt aus.
3. Wählen Sie im Navigationsbereich Overview (Übersicht) aus.
4. Sehen Sie sich die Projektaufgaben an, die Ihnen zugewiesen und von Ihnen erstellt wurden.
  - Die Liste Mitglieder anzeigen + Alle anzeigen, um eine Liste der Projektmitglieder zu sehen.
  - Auf der Karte Repositorien finden Sie die Quell-Repositorys, die dem Projekt zugeordnet sind.
  - Sehen Sie sich die Karte Workflow-Durchläufe an, um die Workflows zu sehen, die dem Projekt zugeordnet sind.
  - Auf der Karte „Pull-Requests öffnen“ finden Sie eine Zusammenfassung des Status des Code-Repositorys sowie der Ihnen zugewiesenen und von Ihnen erstellten Pull Requests.
  - Auf der Karte Meine Entwicklungsumgebungen finden Sie eine Zusammenfassung der Entwicklungsumgebungen, die dem Projekt zugeordnet sind.
  - Auf der Karte Probleme finden Sie eine Zusammenfassung der Ihnen zugewiesenen Aufgaben oder der von Ihnen erstellten Aufgaben.

## Alle Projekte ansehen

In der Projektliste für deinen Bereich kannst du alle Projekte einsehen, für die du über Berechtigungen verfügst.

Verwenden Sie die Konsole, um eine Zusammenfassung der Projektaufgaben, wie z. B. offene Issues und Pull Requests, die Ihnen zugewiesen oder von Ihnen erstellt wurden, sowie der zugehörigen Entwicklungsumgebungen des Projekts anzuzeigen.

Um ein Projekt anzeigen zu können, müssen Sie Mitglied des Projekts sein oder die Rolle eines Space-Administrators für den Bereich besitzen.

Um deine Quell-Repositorys, Workflow-Ausführungen, Issues, Pull Requests, Entwicklungsumgebungen und Issues einzusehen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit dem Projekt, das Sie anzeigen möchten. Wählen Sie unter Projekte Ihr Projekt aus.
3. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
4. Zeigen Sie den Projektnamen, den Pfad, die Projekt-ID und die Beschreibung an.

## Projekteinstellungen anzeigen

In den Projekteinstellungen können Sie Projektmitglieder, Quell-Repositorys, Workflow-Ausführungen, offene Pull-Requests, Projektentwicklungsumgebungen und Probleme anzeigen.

Verwenden Sie die Konsole, um eine Zusammenfassung der Projektaufgaben, wie z. B. offene Issues und Pull Requests, die Ihnen zugewiesen oder von Ihnen erstellt wurden, sowie der zugehörigen Entwicklungsumgebungen des Projekts anzuzeigen.

Um deine Quell-Repositorys, Workflow-Läufe, Issues, Pull Requests, Entwicklungsumgebungen und Issues anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit dem Projekt, das Sie anzeigen möchten. Wählen Sie unter Projekte Ihr Projekt aus.
3. Wählen Sie im Navigationsbereich Projekteinstellungen aus.

4. Zeigen Sie den Projektnamen, den Pfad, die Projekt-ID und die Beschreibung an.

## Zu einem anderen Projekt wechseln in CodeCatalyst

Um zu einem anderen Projekt zu wechseln, wählen Sie in der Konsole aus einer Liste von Projekten, auf die Sie Zugriff haben.

Um zu einem anderen Projekt zu wechseln

1. Wählen Sie in der CodeCatalyst Konsole oben die Projektauswahl aus.
2. Erweitern Sie das Drop-down-Menü und wählen Sie das Projekt aus, zu dem Sie navigieren möchten.

## Löschen eines Projekts in Amazon CodeCatalyst

Sie können ein Projekt löschen, um jeglichen Zugriff auf die Ressourcen des Projekts zu entfernen. Sie benötigen die Rolle Space-Administrator oder Projektadministrator, um ein Projekt löschen zu können. Sobald Sie ein Projekt gelöscht haben, können Projektmitglieder nicht mehr auf Projektressourcen zugreifen, und alle Workflows, die von Quell-Repositorys von Drittanbietern angefordert werden, werden gestoppt.

Um Ihr Projekt zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit dem Projekt, das Sie anzeigen möchten. Wählen Sie unter Projekte Ihr Projekt aus.
3. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
4. Wählen Sie Delete project (Projekt löschen).
5. Geben Sie **delete** die Eingabetaste ein, um den Löschvorgang zu bestätigen.
6. Wählen Sie Delete project (Projekt löschen).

## Verwaltung von Projektmitgliedern

Sie können die Mitglieder in Ihren Projekten über die CodeCatalyst Amazon-Konsole verwalten. Sie können Benutzer hinzufügen oder entfernen, die Rollen aktueller Mitglieder verwalten, Einladungen

zur Teilnahme an Ihrem Projekt versenden und Einladungen stornieren, die noch nicht akzeptiert wurden.

Auf der Mitgliederseite für Space- und Projektbenutzer können Benutzer mehrere Rollen haben. Benutzern mit mehreren Rollen wird ein Indikator angezeigt, wenn sie mehrere Rollen haben, und sie werden mit der Rolle mit den meisten Berechtigungen zuerst angezeigt.

## Mitglieder in einem Projekt anzeigen

Wenn Sie Ihrem Projekt einen Benutzer hinzufügen, weisen Sie eine Rolle zu, die Projektberechtigungen wie folgt gewährt:

- Die Rolle des Projektadministrators verfügt über alle Berechtigungen in einem Projekt. Weisen Sie diese Rolle nur Benutzern zu, die alle Aspekte eines Projekts verwalten müssen, einschließlich der Bearbeitung von Projekteinstellungen, der Verwaltung von Projektberechtigungen und dem Löschen des Projekts. Weitere Informationen finden Sie unter [Rolle als Projektadministrator](#).
- Die Rolle Mitwirkender verfügt über die Berechtigungen, die für die Arbeit an einem Projekt erforderlich sind. Weisen Sie diese Rolle den Benutzern zu, die mit Code, Workflows, Problemen und Aktionen in einem Projekt arbeiten müssen. Weitere Informationen finden Sie unter [Rolle „Mitwirkender“](#).
- Die Prüferrolle verfügt über Überprüfungsberechtigungen. Details hierzu finden Sie unter [Rolle des Prüfers](#).
- Die Rolle „Nur lesen“ hat Leseberechtigungen. Details hierzu finden Sie unter [Rolle „Nur lesen“](#).

Sie müssen keinen Benutzer mit der Rolle Space-Administrator zu Ihrem Projekt einladen, da dieser bereits impliziten Zugriff auf alle Projekte im Space hat.

Wenn Sie einen Benutzer zu Ihrem Projekt einladen (ohne ihm die Rolle des Space-Administrators zuzuweisen), wird der Benutzer in der Tabelle Projektmitglieder unter Projekte und in der Tabelle Projektmitglieder unter Spaces angezeigt.

Um Benutzer und Rollen in Ihrem Bereich anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit dem Projekt, das Sie anzeigen möchten. Wählen Sie unter Projekte Ihr Projekt aus.
3. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.

#### 4. Wählen Sie die Registerkarte Mitglieder.

In der Tabelle Projektmitglieder werden alle Mitglieder angezeigt, die eine Rolle in einem Projekt spielen.

##### Tip

Wenn Sie die Rolle eines Space-Administrators haben, können Sie sehen, zu welchen Projekten Sie direkt eingeladen wurden. Navigieren Sie zu den Projekteinstellungen für das Projekt und wählen Sie dann Meine Projekte aus.

In der Tabelle Space-Administratoren werden Benutzer mit der Rolle Space-Administrator angezeigt. Diese Benutzer werden automatisch (implizit) allen Projekten im Space zugewiesen und haben keine Rolle in einem Projekt.

In der Spalte Status sind die folgenden Werte gültig:

- **Eingeladen** — Die Einladung wurde CodeCatalyst gesendet, aber der Benutzer hat sie noch nicht angenommen oder abgelehnt.
- **Mitglied** — Der Benutzer hat die Einladung angenommen.

#### Themen

- [Einen Benutzer zu Ihrem Projekt einladen](#)
- [Eine Einladung stornieren](#)
- [Einen Benutzer aus Ihrem Projekt entfernen](#)
- [Eine Einladung für ein Projekt annehmen oder ablehnen](#)

## Einen Benutzer zu Ihrem Projekt einladen

Sie können die Konsole verwenden, um Benutzer zu Ihrem Projekt einzuladen. Sie können Mitglieder Ihres Bereichs einladen oder Namen von außerhalb Ihres Bereichs hinzufügen.

Um Benutzer zu Ihrem Projekt einzuladen, müssen Sie mit der Rolle Projektadministrator oder Space-Administrator angemeldet sein.

Sie müssen keinen Benutzer mit der Rolle eines Space-Administrators zu Ihrem Projekt einladen, da dieser bereits impliziten Zugriff auf alle Projekte im Space hat.

Wenn Sie einen Benutzer zu Ihrem Projekt einladen (ohne ihm die Rolle des Space-Administrators zuzuweisen), wird der Benutzer in der Tabelle Projektmitglieder unter Projekte und in der Tabelle Projektmitglieder unter Spaces angezeigt.

Um ein Mitglied über den Tab Projekteinstellungen zu Ihrem Projekt einzuladen

1. Navigiere zu deinem Projekt.


 Tip

In der oberen Navigationsleiste können Sie auswählen, welches Projekt angezeigt werden soll.

2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Wählen Sie die Registerkarte Mitglieder.
4. Wählen Sie unter Projektmitglieder die Option Neues Mitglied einladen aus.
5. Geben Sie die E-Mail-Adresse des neuen Mitglieds ein, wählen Sie die Rolle für dieses Mitglied aus und wählen Sie dann Einladen aus. Weitere Informationen zu Rollen finden Sie unter [Mitrollen in Amazon arbeiten CodeCatalyst](#).

Um ein Mitglied von der Projektübersichtsseite aus zu Ihrem Projekt einzuladen

1. Navigiere zu deinem Projekt.

 Tip

In der oberen Navigationsleiste können Sie auswählen, welches Projekt angezeigt werden soll.

2. Wählen Sie die Schaltfläche Mitglieder +.
3. Geben Sie die E-Mail-Adresse des neuen Mitglieds ein, wählen Sie die Rolle für dieses Mitglied aus und wählen Sie dann Einladen aus. Weitere Informationen zu Rollen finden Sie unter [Mitrollen in Amazon arbeiten CodeCatalyst](#).

## Eine Einladung stornieren

Wenn Sie kürzlich eine Einladung gesendet haben, können Sie sie stornieren, solange die Einladung noch nicht angenommen wurde.

Um Projekteinladungen verwalten zu können, benötigen Sie die Rolle Projektadministrator oder Space-Administrator.

Um die Einladung eines Projektmitglieds zu stornieren

1. Navigieren Sie zu dem Projekt, an das Sie eine Einladung gesendet haben, die Sie stornieren möchten.
2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Rufen Sie die Registerkarte Mitglieder auf und überprüfen Sie, ob das Mitglied den Status Eingeladen hat.

### Note

Sie können nur eine Einladung stornieren, die noch nicht angenommen wurde.

4. Wählen Sie die Option neben der Zeile mit dem eingeladenen Mitglied und wählen Sie dann Einladung stornieren aus.
5. Ein Bestätigungsfenster wird angezeigt. Wählen Sie zur Bestätigung „Einladung stornieren“.

## Einen Benutzer aus Ihrem Projekt entfernen

Sie können die Konsole verwenden, um einen Benutzer aus Ihrem Projekt zu entfernen.

Um einen Benutzer aus Ihrem Projekt zu entfernen, müssen Sie mit der Rolle Projektadministrator oder Space-Administrator angemeldet sein.

### Note

Wenn Sie einen Benutzer aus allen Projekten innerhalb eines Bereichs entfernen, wird der Benutzer automatisch aus diesem Bereich entfernt.

## Um einen Benutzer aus einem Projekt zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich mit dem Projekt, das Sie anzeigen möchten. Wählen Sie unter Projekte Ihr Projekt aus.
3. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
4. Wählen Sie die Registerkarte Mitglieder.
5. Wählen Sie den Selektor neben dem Profil aus, das Sie entfernen möchten, und wählen Sie dann Entfernen aus.
6. Bestätigen Sie, dass Sie den Benutzer entfernen möchten, und wählen Sie dann Entfernen.

## Eine Einladung für ein Projekt annehmen oder ablehnen

Möglicherweise erhalten Sie eine E-Mail-Einladung zur Teilnahme an einem CodeCatalyst Amazon-Projekt. Sie können die Einladung akzeptieren oder ablehnen.

### Akzeptieren oder Ablehnen einer Einladung

1. Öffnen Sie die Einladungs-E-Mail.
2. Wählen Sie den Projektlink in der E-Mail aus.
3. Wählen Sie Annehmen oder Ablehnen.

Wenn Sie Ablehnen wählen, wird eine E-Mail an das Projektmanagementkonto gesendet, in der es darüber informiert wird, dass Sie die Einladung abgelehnt haben.

## Teams für Projekte leiten

Nachdem Sie ein Projekt erstellt haben, können Sie Teams hinzufügen. Teams ermöglichen es Ihnen, Benutzer so zu gruppieren, dass sie CodeCatalyst als Projekt- und Bereichsmitglieder Berechtigungen gemeinsam nutzen und Projekte, Problemverfolgung, Rollen und Ressourcen verwalten können.

Sie benötigen die Rolle eines Projektadministrators, um Teams für Ihr Projekt verwalten zu können.

### Themen

- [Ein Team zu einem Projekt hinzufügen](#)



- [Eine Projektrolle für ein Team verwalten](#)
- [Eine Projektrolle für ein Team entfernen](#)

## Ein Team zu einem Projekt hinzufügen

Sie können Teams verwalten, in denen die Teammitglieder auf Ressourcen in Ihrem Projekt zugreifen können.

Auf der Mitgliederseite für Space- und Projektbenutzer können Benutzer mehrere Rollen haben. Benutzern mit mehreren Rollen wird ein Indikator angezeigt, wenn sie mehrere Rollen haben, und sie werden mit der Rolle mit den meisten Berechtigungen zuerst angezeigt.

Um ein Team hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt. Wählen Sie Projekteinstellungen und dann Teams aus.
3. Wählen Sie Team hinzufügen aus.
4. Wählen Sie unter Team ein Team aus der Liste der verfügbaren Teams aus.
5. Wählen Sie unter Projektrollen eine Rolle aus der Liste der verfügbaren Projektrollen in aus CodeCatalyst.
  - Projektadministrator — Einzelheiten finden Sie unter [Rolle als Projektadministrator](#).
  - Mitwirkender — Einzelheiten finden Sie unter [Rolle „Mitwirkender“](#).
  - Prüfer — Einzelheiten finden Sie unter [Rolle des Prüfers](#).
  - Schreibgeschützt — Einzelheiten finden Sie unter [Rolle „Nur lesen“](#).
6. Wählen Sie Team hinzufügen aus.

## Eine Projektrolle für ein Team verwalten

Ein Team kann über Rollenberechtigungen, wie z. B. Hauptbenutzer, in einem Bereich verfügen. Sie können die Space-Rolle für ein Team ändern, beachten Sie jedoch, dass alle Mitglieder des Teams diese Berechtigungen erben.

Um eine Projektrolle hinzuzufügen oder zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

2. Navigiere zu deinem Bereich. Wählen Sie Projekteinstellungen und dann Teams aus.
3. Um eine Rolle zu ändern, wählen Sie den Selektor neben dem Team in dieser Liste aus und wählen Sie dann Rolle ändern aus. Um eine Rolle hinzuzufügen, wählen Sie Projektrolle hinzufügen aus. Wählen Sie unter Projekt das Projekt aus, das Sie hinzufügen möchten, und wählen Sie unter Rolle die Rolle aus. Wählen Sie eine der verfügbaren Projektrollen aus:
  - Projektadministrator — Einzelheiten finden Sie unter [Rolle als Projektadministrator](#).
  - Mitwirkender — Einzelheiten finden Sie unter [Rolle „Mitwirkender“](#).
  - Prüfer — Einzelheiten finden Sie unter [Rolle des Prüfers](#).
  - Schreibgeschützt — Einzelheiten finden Sie unter [Rolle „Nur lesen“](#).
4. Wählen Sie Speichern.

## Eine Projektrolle für ein Team entfernen

CodeCatalystIn können Sie die Projektrollen für Ihr Team einsehen. Sie können auch die Mitglieder eines Teams anzeigen. Sie können die Projektrolle für ein Team entfernen.

Um eine Projektrolle zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem Bereich. Wählen Sie Projekteinstellungen und dann Teams aus.
3. Wählen Sie die Registerkarte Projektrollen aus.
4. Wählen Sie die Rolle aus, die Sie entfernen möchten.

### Important

Wenn Sie eine Rolle aus einem Team entfernen, werden die zugehörigen Berechtigungen für alle Benutzer im Team entfernt.

5. Klicken Sie auf Speichern.

## Verwaltung von Maschinenressourcen

Eine Maschinenressource stellt Ihre Identität gegenüber Ihrer autorisierten Ressource dar, wenn Sie CodeCatalyst über SSO darauf zugreifen. Maschinenressourcen werden

verwendet, um Berechtigungen für Ressourcen in Ihrem Projekt zu erteilen, z. B. Blueprints und Workflows. Sie können die Maschinenressourcen in Ihrem Projekt anzeigen und wählen, ob Sie Maschinenressourcen für Ihr Projekt aktivieren oder deaktivieren möchten. Beispielsweise möchten Sie möglicherweise eine Maschinenressource deaktivieren, um den Zugriff zu verwalten, und sie dann später wieder aktivieren.

Diese Operationen sind für Maschinenressourcen verfügbar, wenn eine Maschinenressource gesperrt oder deaktiviert werden muss. Wenn Sie beispielsweise vermuten, dass Anmeldeinformationen kompromittiert wurden, können Sie die Computerressource deaktivieren. Im Allgemeinen müssen diese Operationen nicht verwendet werden.

Sie benötigen die Rolle eines Space-Administrators oder eines Projektadministrators, um diese Seite anzeigen und Maschinenressourcen auf Projektebene verwalten zu können.

Themen

- [Maschinenressourcen anzeigen](#)
- [Deaktivieren Sie die Maschinenressourcen](#)
- [Aktivieren Sie Maschinenressourcen](#)

## Maschinenressourcen anzeigen

Sie können eine Liste der Maschinenressourcen anzeigen, die in Ihrem Projekt verwendet werden.

Sie müssen über die Rolle eines Space-Administrators oder eines Projektadministrators verfügen.

Um Maschinenressourcen anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt und wählen Sie dann Projekteinstellungen aus. Wählen Sie Maschinenressourcen aus.
3. Wählen Sie in der Dropdownliste Workflow-Aktion aus, um nur die Maschinenressourcen für Workflows anzuzeigen. Wählen Sie Blueprint aus, um nur die Maschinenressourcen für Blueprints anzuzeigen.

Sie können mithilfe des Felds Filter auch nach einem Namen filtern.

## Deaktivieren Sie die Maschinenressourcen

Sie können wählen, ob Sie Maschinenressourcen deaktivieren möchten, die in Ihrem Projekt verwendet werden.

### Important

Durch das Deaktivieren von Maschinenressourcen werden alle Berechtigungen für alle zugehörigen Blueprints oder Workflows im Bereich entfernt.

Sie müssen über die Rolle eines Space-Administrators oder eines Projektadministrators verfügen.

Um Maschinenressourcen zu deaktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt und wählen Sie dann Projekteinstellungen aus. Wählen Sie Maschinenressourcen aus.
3. Wählen Sie eine der folgenden Optionen aus.

### Important

Durch das Deaktivieren von Maschinenressourcen werden alle Berechtigungen für alle zugehörigen Blueprints oder Workflows im Bereich entfernt.

- Um sie einzeln zu deaktivieren, wählen Sie den Selektor neben einer oder mehreren Maschinenressourcen aus, die Sie deaktivieren möchten. Wählen Sie Deaktivieren und anschließend Diese Ressource aus.
- Um alle Ressourcen zu deaktivieren, wählen Sie „Deaktivieren“ und anschließend „Alle Ressourcen“.
- Um alle Workflow-Aktionen zu deaktivieren, wählen Sie Deaktivieren und dann Alle Workflow-Aktionen aus.
- Um alle Blueprints zu deaktivieren, wählen Sie Deaktivieren und dann Alle Blueprints aus.

## Aktivieren Sie Maschinenressourcen

Sie können wählen, ob Maschinenressourcen aktiviert werden sollen, die in Ihrem Projekt verwendet werden und die deaktiviert wurden.

Sie müssen über die Rolle eines Space-Administrators oder eines Projektadministrators verfügen.

Um Maschinenressourcen zu aktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt und wählen Sie dann Projekteinstellungen aus. Wählen Sie Maschinenressourcen aus.
3. Wählen Sie eine der folgenden Optionen aus.
  - Um sie einzeln zu aktivieren, wählen Sie den Selektor neben einer oder mehreren Maschinenressourcen, die Sie aktivieren möchten. Wählen Sie Aktivieren und anschließend Diese Ressource aus.
  - Um alle Ressourcen zu aktivieren, wählen Sie Aktivieren und dann Alle Ressourcen aus.
  - Um alle Workflow-Aktionen zu aktivieren, wählen Sie Aktivieren und dann Alle Workflow-Aktionen aus.
  - Um alle Blueprints zu aktivieren, wählen Sie Aktivieren und dann Alle Blueprints aus.

## Kontingente für Projekte in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Limits für Projekte bei Amazon beschrieben CodeCatalyst. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

Maximale Anzahl von Projekten pro Bereich	100
Maximale Anzahl von Projekten, zu denen ein Benutzer gehören kann	1.000
Maximale Anzahl von Mitgliedern, die zu einem Projekt gehören können.	10.000

Projektnamen	<p>Projektnamen müssen innerhalb eines Leerzeichens eindeutig sein. Namen müssen zwischen 3 und 63 Zeichen lang sein. Bei den Namen muss die Groß- und Kleinschreibung beachtet werden. Projektnamen müssen mit einem alphanumerischen Zeichen beginnen. Zulässige Zeichen: A-Z, a-z, 0-9, Leerzeichen und., _ (Unterstrich) — (Bindestrich)</p> <p>Projektnamen dürfen keines der folgenden Zeichen enthalten: ! ? @ # \$ % ^ &amp; * ( ) + = { } [ ]   \ / &gt; &lt; ~ ` ' " ; :</p>
Projektbeschreibungen	<p>Projektbeschreibungen können bis zu 200 Zeichen lang sein. Zulässige Zeichen: A-Z, a-z, 0-9, Leerzeichen und., _ (Unterstrich) — (Bindestrich). Projektbeschreibungen sind optional.</p>

## Arbeiten mit Benachrichtigungen in CodeCatalyst

Sie können Benachrichtigungen einrichten, um Ihre Projekte und Ressourcen in zu überwachen CodeCatalyst. Benutzer können in jedem Projekt, in dem sie Mitglied sind, die Projektereignisse auswählen, über die sie E-Mails erhalten möchten. Du kannst dich auch dafür entscheiden, Benachrichtigungen, die an ein ganzes Team gesendet werden, in einer Team-Messaging-Anwendung wie Slack zu konfigurieren, indem du den Zugriff zwischen einem CodeCatalyst Space und einem Slack-Workspace konfigurierst und dann Benachrichtigungen für ein Projekt so konfigurierst, dass sie an einen oder mehrere Kanäle in diesem Slack-Workspace gesendet werden. Sobald du den Zugriff zwischen einem CodeCatalyst Space und einem Slack-Workspace konfiguriert hast, haben Projektmitglieder auch die Möglichkeit, ihre eigenen Slack-Mitglieds-IDs hinzuzufügen, sodass sie direkt über CodeCatalyst Ereignisse in verbundenen Slack-Workspaces und -Kanälen informiert werden können.

**Note**

Bei den Projektereignissen, die an Slack gesendet werden können, handelt es sich nicht um dieselbe Gruppe von Ereignissen, über die Benutzer per E-Mail benachrichtigt werden können.

**Themen**

- [Wie funktionieren Benachrichtigungen?](#)
- [Erste Schritte mit Slack-Benachrichtigungen](#)
- [Benachrichtigungen in Amazon verwalten CodeCatalyst](#)

## Wie funktionieren Benachrichtigungen?

Du kannst dein Projekt so einrichten, dass Benachrichtigungen an deine Team-Messaging-Anwendung wie Slack gesendet werden.

### Welche Berechtigungen sind für Benachrichtigungen erforderlich?

Jedes Projektmitglied kann die Benachrichtigungseinstellungen für einen Kanal in konfigurieren, anzeigen, aktualisieren oder löschen CodeCatalyst. Allerdings können nur Benutzer mit der Rolle Space-Administrator Slack-Workspaces hinzufügen oder löschen. Alle Benutzer können konfigurieren, über welche Projektereignisse sie E-Mails für die Projekte erhalten möchten, zu denen sie gehören. CodeCatalyst

### Über welche CodeCatalyst Ereignisse kann ich Benachrichtigungen konfigurieren?

Du kannst so konfigurieren CodeCatalyst , dass Benachrichtigungen über Workflow-Ereignisse an einen oder mehrere Slack-Channels gesendet werden. Sobald Benachrichtigungen zwischen einem CodeCatalyst Projekt und Slack konfiguriert wurden, können Projektbenutzer wählen, ob sie ihre eigene Slack-Mitglieds-ID hinzufügen möchten, um in Slack-Kanälen Direktnachrichten über Ereignisse zu erhalten. CodeCatalyst Nutzer, die ihre Slack-Mitglieds-IDs hinzufügen, erhalten direkte Erwähnungen zu ihren IDs in den für ihre Projekte konfigurierten Slack-Channels, was dazu beiträgt, auf Ereignisse aufmerksam zu machen, die ihnen wichtig sind.

Du kannst auch wählen, über welche Ereignisse du E-Mails erhalten möchtest. Diese E-Mails werden an die für Ihre AWS Builder-ID konfigurierte E-Mail-Adresse gesendet.

## Wie werden Benachrichtigungen angezeigt?

Du kannst so konfigurieren CodeCatalyst , dass Benachrichtigungen an einen oder mehrere Slack-Channels gesendet werden. Du musst autorisieren, Berechtigungen für CodeCatalyst den Zugriff auf deinen Slack-Workspace zu erteilen. Sobald die Autorisierung erteilt wurde, CodeCatalyst kann er Benachrichtigungen an die von dir konfigurierten Slack-Channels senden. Wenn ein Projektmitglied sich dafür entscheidet, seine Slack-Mitglieds-ID hinzuzufügen, kann es Erwähnungen über CodeCatalyst Ereignisse in den für dieses Projekt konfigurierten Slack-Channels erhalten.

## Wie richte ich Benachrichtigungen ein?

E-Mail-Benachrichtigungen werden als Teil von konfiguriert CodeCatalyst. Projektbenutzer können auf ihrer Seite Meine Einstellungen auswählen, über welche Ereignisse sie E-Mails erhalten möchten.

Um Slack-Benachrichtigungen für deine Projektressourcen einzurichten, musst du die folgenden übergeordneten Aufgaben erledigen.

Um Benachrichtigungen einzurichten (Aufgaben auf hoher Ebene)

1. CodeCatalystIn richten Sie eine Verbindung zwischen CodeCatalyst und einem Messaging-Client wie Slack ein. Sobald ein Slack-Workspace verbunden ist, steht er allen Projekten im Space zur Verfügung.

### Note

Nur Benutzer mit einer Space-Administratorrolle können einen Slack-Workspace hinzufügen oder löschen.

2. Füge in deinem Projekt den Channel hinzu CodeCatalyst, über den dein Team Benachrichtigungen erhalten soll.
3. In CodeCatalyst aktivierst du Benachrichtigungen für verschiedene Ereignisse, wie z. B. Fehler bei der Workflow-Ausführung, und gibst den Kanal an, an den sie gesendet werden sollen.

Die detaillierten Schritte finden Sie unter [Erste Schritte mit Slack-Benachrichtigungen](#).

Sobald Benachrichtigungen zwischen einem CodeCatalyst Space und Slack konfiguriert wurden, können Benutzer wählen, ob sie ihre eigenen Slack-Mitglieds-IDs hinzufügen möchten, um Direktnachrichten über CodeCatalyst Ereignisse in den für ihre Projekte konfigurierten Slack-Kanälen zu erhalten.



## Erste Schritte mit Slack-Benachrichtigungen

Nachdem du ein Projekt erstellt hast, kannst du Slack-Benachrichtigungen einrichten, die deinem Team helfen, die Projektressourcen zu überwachen.

Diese Schritte führen dich durch die Einrichtung von Slack-Benachrichtigungen zum ersten Mal in CodeCatalyst. Wenn du Benachrichtigungen bereits konfiguriert hast, findest du weitere Informationen unter [Benachrichtigungen in Amazon verwalten CodeCatalyst](#).

### Note

Bei den Projektereignissen, die an Benachrichtigungskanäle gesendet werden können, handelt es sich nicht um dieselbe Gruppe von Ereignissen, über die Benutzer per E-Mail benachrichtigt werden können. Weitere Informationen finden Sie unter [Benachrichtigungen in Amazon verwalten CodeCatalyst](#).

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Connect dich mit CodeCatalyst deinem Slack-Workspace](#)
- [Schritt 2: Füge deinen Slack-Kanal hinzu CodeCatalyst](#)
- [Schritt 3: Teste Benachrichtigungen von bis CodeCatalyst zu Slack](#)
- [Schritt 4: Nächste Schritte](#)

### Voraussetzungen

Bevor Sie beginnen, muss Folgendes sichergestellt sein:

- Ein CodeCatalyst Leerzeichen. Informationen zum Erstellen eines CodeCatalyst Bereichs und zur ersten Anmeldung finden Sie unter [Einrichten CodeCatalyst](#).
- Ein CodeCatalyst Projekt. Weitere Informationen finden Sie unter [Ein Projekt in Amazon erstellen CodeCatalyst](#).
- Ein CodeCatalyst Konto mit der Rolle Projektadministrator oder Space-Administrator. Weitere Informationen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).
- Ein Slack-Konto und ein Slack-Workspace, auf den zugegriffen werden kann von. CodeCatalyst

- Ein Slack-Kanal, über den Benachrichtigungen gesendet CodeCatalyst werden. Der Kanal kann öffentlich oder privat sein.

## Schritt 1: Connect dich mit CodeCatalyst deinem Slack-Workspace

Nur Benutzer mit der Space-Administratorrolle können Slack-Workspaces hinzufügen oder löschen. Das Hinzufügen oder Löschen eines Slack-Workspace wirkt sich auf alle Projekte im Space aus. Führe einen sicheren OAuth-Authentifizierungs-Handshake mit deinem Slack-Workspace CodeCatalyst durch, um die Verbindung zwischen CodeCatalyst und Slack herzustellen.

Verwende die folgenden Anweisungen, um eine Verbindung zu deinem Slack-Workspace herzustellen CodeCatalyst .

### Note

Dies muss nur einmal für jeden Slack-Workspace gemacht werden. Du kannst dann Benachrichtigungen für jeden Slack-Kanal einrichten.

Um dich mit deinem Slack-Workspace CodeCatalyst zu verbinden

1. Öffne die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt.
3. Wählen Sie im Navigationsbereich Projekteinstellungen aus.
4. Wählen Sie die Registerkarte Benachrichtigungen.
5. Wählen Sie Benachrichtigungen konfigurieren.
6. Wähle Mit dem Slack-Workspace verbinden.
7. Lies den Inhalt des Dialogfelds und wähle dann Mit dem Slack-Workspace verbinden.
8. In der AWSChatbot-Nachricht:
  - a. Wähle oben rechts den Slack-Workspace aus, der deinen Channel enthält.
  - b. Wählen Sie Allow.

Du kehrst zur CodeCatalyst Konsole zurück.

9. Fahren Sie fort mit [Schritt 2: Füge deinen Slack-Kanal hinzu CodeCatalyst](#).

## Schritt 2: Füge deinen Slack-Kanal hinzu CodeCatalyst

Du benötigst die Slack-Kanal-ID, zu der du deinen Kanal hinzufügen möchtest. CodeCatalyst

Um deine Slack-Kanal-ID zu erhalten

1. Melde dich bei Slack an. Weitere Informationen findest du unter [Bei Slack anmelden](#).
2. Gehe zum Slack-Workspace, der den Channel enthält, in den du Benachrichtigungen senden möchtest. Weitere Informationen findest du unter [Zwischen Slack-Workspaces wechseln](#) oder [In weiteren Slack-Workspaces anmelden](#).
3. Öffne im Navigationsbereich das Kontextmenü (Rechtsklick) für den Channel, in den du Benachrichtigungen senden möchtest, und wähle Channel-Details öffnen aus.

Die Kanal-ID wird unten im Dialogfeld angezeigt.

4. Kopieren Sie den Kanal-ID-Wert. Sie benötigen ihn im nächsten Schritt.

Mit der Kanal-ID, die du gerade kopiert hast, kannst du jetzt deinen Slack-Kanal mit verbinden.  
CodeCatalyst

Um deinen Slack-Kanal hinzuzufügen CodeCatalyst

1. Bevor du anfängst, füge die AWS Chatbot-App wie folgt zum Channel hinzu, falls dein Slack-Kanal privat ist:
  - a. Gib im Nachrichtefeld deines Slack-Kanals die aws-App ein **@aws** und wähle sie aus dem Dialogfeld aus.
  - b. Drücken Sie die Eingabetaste.

Eine Slackbot-Nachricht wird angezeigt, die darauf hinweist, dass sich der AWS Chatbot nicht im privaten Channel befindet.

- c. Wähle Invite Them, um den AWS Chatbot in den Channel einzuladen.
2. Wählen Sie in der CodeCatalyst Konsole „Weiter“.
  3. Füge in Channel-ID die Slack-Channel-ID ein, die du zuvor erhalten hast.
  4. Gib im Feld Kanalname einen Namen ein. Wir empfehlen, den Namen des Slack-Channels zu verwenden.
  5. Wählen Sie Weiter.

6. Wähle unter Benachrichtigungsereignisse auswählen den Ereignistyp aus, für den du Benachrichtigungen erhalten möchtest.
7. Wählen Sie Finish (Abschließen).

### Schritt 3: Teste Benachrichtigungen von bis CodeCatalyst zu Slack

Nachdem dein Projekt so konfiguriert wurde, dass es Benachrichtigungen zum Workflow-Status sendet, kannst du deine Benachrichtigungen in Slack einsehen.

Um deine Benachrichtigungen in Slack einzusehen

1. [Starte in deinem CodeCatalyst Projekt manuell einen Workflow](#), um einen Workflow-Lauf abzuschließen und eine Statusbenachrichtigung zu erhalten, wenn der Lauf abgeschlossen ist.
2. Sieh dir in Slack den Channel an, den du für Benachrichtigungen eingerichtet hast. Deine Benachrichtigungen zeigen den aktuellen Status jeder Workflow-Ausführung an und ob sie fehlgeschlagen oder erfolgreich war.

### Schritt 4: Nächste Schritte

Sobald ein Slack-Workspace für deinen CodeCatalyst Bereich konfiguriert ist, kannst du weitere Slack-Channels, bestehende CodeCatalyst Projekte hinzufügen und sie für neue Projekte hinzufügen, nachdem du sie erstellt hast. Du kannst Projektnutzer auch darüber informieren, dass sie persönliche Slack-Benachrichtigungen für ihre Slack-Mitglieds-IDs konfigurieren und die Ereignisse konfigurieren können, für die sie E-Mails erhalten. Weitere Informationen finden Sie unter [Benachrichtigungen in Amazon verwalten CodeCatalyst](#).

## Benachrichtigungen in Amazon verwalten CodeCatalyst

Sie können so konfigurieren CodeCatalyst , dass Benachrichtigungen über Ereignisse in Ihrem Projekt gesendet werden. Du kannst Benachrichtigungen an Messaging-Clients wie Slack-Channels senden. Projektbenutzer können wählen, über welche Projekt Ereignisse sie per E-Mail an die für ihr Profil konfigurierte E-Mail-Adresse benachrichtigt werden.

**Note**

Bei den Projektereignissen, die an Benachrichtigungskanäle gesendet werden können, handelt es sich nicht um dieselbe Gruppe von Ereignissen, über die Benutzer per E-Mail benachrichtigt werden können.

**Themen**

- [Verwaltung von Benachrichtigungen, die direkt an Sie gesendet werden](#)
- [An Channels gesendete Benachrichtigungen verwalten](#)

**Verwaltung von Benachrichtigungen, die direkt an Sie gesendet werden**

Du kannst wählen, ob du E-Mail-Benachrichtigungen über Ereignisse in jedem Projekt erhalten möchtest, in dem du Mitglied bist. Diese E-Mails werden an die in Ihrer AWS Builder-ID konfigurierte E-Mail-Adresse gesendet. Standardmäßig erhalten Sie E-Mails über alle Projektereignisse, für die E-Mails gesendet werden können.

Wenn ein Projekt so konfiguriert wurde, dass Benachrichtigungen an einen Slack-Kanal gesendet werden, kannst du deine Slack-Mitglieds-ID hinzufügen, um direkte Erwähnungen über CodeCatalyst Ereignisse in diesem Slack-Kanal zu erhalten. Das kann dir helfen, auf Ereignisse aufmerksam zu machen, die in den Projekten stattfinden, in denen du eine Rolle spielst.

Um E-Mail-Benachrichtigungen für Projektereignisse zu konfigurieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie in der oberen Menüleiste Ihr Profil-Badge und dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.

**Tip**

Sie können Ihr Benutzerprofil auch finden, indem Sie die Mitgliederseite für ein Projekt oder einen Bereich aufrufen und Ihren Namen aus der Mitgliederliste auswählen.

3. Suchen Sie unter E-Mail-Benachrichtigungen das Projekt in der Liste, für das Sie E-Mail-Benachrichtigungen konfigurieren möchten, und wählen Sie Bearbeiten aus.

4. Wählen Sie die Ereignisse aus, für die Sie E-Mails erhalten möchten, und wählen Sie dann Speichern.

Um persönliche Slack-Benachrichtigungen zu konfigurieren

1. Öffne die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie in der oberen Menüleiste Ihr Profil-Badge und dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.

 Tip

Sie können Ihr Benutzerprofil auch finden, indem Sie die Mitgliederseite für ein Projekt oder einen Bereich aufrufen und Ihren Namen aus der Mitgliederliste auswählen.

3. Wähle unter Persönliche Slack-Benachrichtigungen die Option „Slack-ID Connect“ und dann „Mit Slack-Workspace verbinden“. Ein separates Fenster wird geöffnet.

 Tip

Diese Option ist nicht konfigurierbar, es sei denn, ein Benutzer mit der Rolle Space-Administrator hat einen Slack-Workspace für deinen CodeCatalyst Bereich hinzugefügt. Weitere Informationen erhalten Sie unter [Erste Schritte mit Slack-Benachrichtigungen](#) und [An Channels gesendete Benachrichtigungen verwalten](#).

4. Vergewissere dich im Fenster mit der Anforderung von Berechtigungen, dass der Name des Workspace mit dem für den Space konfigurierten Slack-Workspace übereinstimmt. CodeCatalyst Wähle Zulassen, um den AWS Chatbot Zugriff auf den Workspace zu gewähren. Das Fenster wird geschlossen und der Slack-Workspace zeigt den Verbindungsstatus als Verbunden an.

 Tip

Wenn sich der Verbindungsstatus nicht ändert, überprüfe, ob beim Verbinden des Slack-Workspace ein Fehler aufgetreten ist. Möglicherweise musst du nach oben scrollen, um den Fehler zu sehen.

5. Um keine persönlichen Slack-Benachrichtigungen mehr zu erhalten, wähle den verbundenen Slack-Workspace und dann Disconnect Slack ID aus.

## An Channels gesendete Benachrichtigungen verwalten

Du kannst wählen, ob du Benachrichtigungen über Projektereignisse hinzufügen und verwalten möchtest, die an Teamressourcen CodeCatalyst gesendet werden, z. B. an einen Team-Slack-Kanal. Auf diese Weise können Sie sicherstellen, dass Ihr gesamtes Team über wichtige Ereignisse informiert ist, z. B. wenn eine Workflow-Ausführung fehlschlägt.

### Note

Jedes Mitglied eines Projekts kann Benachrichtigungen verwalten, die an Kanäle für dieses Projekt gesendet werden. Allerdings können nur Benutzer mit der Rolle Space-Administrator Slack-Workspaces hinzufügen oder löschen.

### Einen Benachrichtigungskanal für ein Projekt hinzufügen

Du kannst einen Channel hinzufügen, über den du Benachrichtigungen erhalten möchtest, z. B. den Slack-Kanal deines Teams.

Um einen Slack-Kanal für Benachrichtigungen hinzuzufügen

1. Wenn du deinen ersten Slack-Kanal hinzufügst, sieh stattdessen nach. [Erste Schritte mit Slack-Benachrichtigungen](#)

Nachdem du deinen ersten Channel eingerichtet hast, kehre zu diesem Verfahren zurück, um weitere Channels einzurichten.

2. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Navigieren Sie zu Ihrem Projekt.
4. Wählen Sie im Navigationsbereich Projekteinstellungen aus.
5. Wählen Sie die Registerkarte Benachrichtigungen.
6. Wählen Sie Add channel (Kanal hinzufügen) aus.
7. Wähle Workspace und wähle dann den Slack-Workspace aus, der den Channel enthält, an den du Benachrichtigungen senden möchtest.

Wenn dein Slack-Workspace nicht in der Liste enthalten ist, kannst du ihn hinzufügen, indem du den Anweisungen unter folgst. [Erste Schritte mit Slack-Benachrichtigungen](#)

8. Wenn der Slack-Channel, den du hinzufügen möchtest, privat ist, führe folgende Schritte durch, bevor du eine Channel-ID eingibst:

- a. Gib im Nachrichtefeld deines Slack-Kanals die aws-App ein **@aws** und wähle sie aus dem Pop-up aus.
- b. Drücken Sie die Eingabetaste.

Eine Slackbot-Nachricht wird angezeigt, die darauf hinweist, dass sich der AWS Chatbot nicht im privaten Channel befindet.

- c. Wähle Invite Them, um AWS Chatbot in den Channel einzuladen.
9. Gib im CodeCatalyst Feld Channel-ID die Channel-ID von Slack ein. Um die ID zu finden, gehe zu Slack und klicke im Navigationsbereich mit der rechten Maustaste auf den Channel und wähle Channel-Details öffnen.

Die Kanal-ID wird unten im Dialogfeld angezeigt.

10. Geben Sie im Feld Kanalname einen Namen ein. Wir empfehlen, den Namen des Slack-Channels zu verwenden.
11. Wähle unter Benachrichtigungsereignisse auswählen den Ereignistyp aus, für den du Benachrichtigungen erhalten möchtest.
12. Wählen Sie Add (Hinzufügen) aus.

## Benachrichtigungen für einen Benachrichtigungskanal bearbeiten

Du kannst ändern, auf welche Kanäle Benachrichtigungen gesendet werden, und du kannst bestimmte Benachrichtigungen ganz ausschalten.

Um Benachrichtigungen zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt.
3. Wählen Sie im Navigationsbereich Projekteinstellungen aus.
4. Wählen Sie die Registerkarte Benachrichtigungen.
5. Wählen Sie Benachrichtigungen bearbeiten.
6. Führen Sie eine der folgenden Aktionen aus:
  - Um eine Benachrichtigung an einen bestimmten Kanal zu senden, wählen Sie den Kanal aus der Drop-down-Liste aus.



- Um eine Benachrichtigung global zu deaktivieren, klicken Sie auf den Schalter neben der Benachrichtigung.
- Um das Senden einer Benachrichtigung an einen bestimmten Kanal zu beenden, wählen Sie das X auf dem Kanal.

7. Wählen Sie Speichern.

## Einen Kanal entfernen

Um einen Kanal zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt. Wählen Sie im Navigationsbereich Projekteinstellungen aus.
3. Wählen Sie auf der Seite mit den Projekteinstellungen die Registerkarte Benachrichtigungen aus.
4. Wählen Sie den Indikator neben dem Kanal aus, den Sie entfernen möchten, und wählen Sie dann Kanal entfernen. Wenn Sie dazu aufgefordert werden, wählen Sie im Bestätigungsfenster Ok.

# Baupläne in CodeCatalyst

Blueprints sind Generatoren für beliebigen Code, die eine architektonische Komponente eines CodeCatalyst Projekts darstellen. Die Komponente kann aus allem bestehen, von einem Workflow in einer einzigen Datei bis hin zum gesamten Projekt mit Beispielcode. Blueprints verwenden einen beliebigen Satz von Optionen und verwenden diese, um einen beliebigen Satz von Ausgabecode zu generieren, der an ein Projekt weitergeleitet wird. Wenn der Blueprint mit den neuesten bewährten Methoden oder neuen Optionen aktualisiert wird, kann er die relevanten Teile Ihrer Codebasis in Projekten, die diesen Blueprint enthalten, neu generieren.

Sie können einen CodeCatalyst Amazon-Blueprint verwenden, um ein vollständiges Projekt mit einem Quell-Repository, Beispielquellcode, CI/CD-Workflows, Build- und Testberichten und integrierten Tools zur Problemverfolgung zu erstellen. Ein CodeCatalyst Blueprint generiert Ressourcen und Quellcode auf der Grundlage der festgelegten Konfigurationsparameter. Wenn Sie einen CodeCatalyst -verwalteten Blueprint verwenden, bestimmt der von Ihnen gewählte Blueprint, welche Ressourcen Ihrem Projekt hinzugefügt werden und welche Tools CodeCatalyst erstellt oder konfiguriert werden, sodass Sie Ihre Projektressourcen verfolgen und verwenden können. Als Blueprint-Benutzer können Sie ein Projekt mit einem Blueprint erstellen oder sie auf ein vorhandenes Projekt anwenden. CodeCatalyst Sie können mehrere Blueprints in Ihrem Projekt anwenden, und jeder kann als unabhängige Komponente angewendet werden. Sie können beispielsweise ein Projekt haben, das mit einem Webanwendungs-Blueprint erstellt wurde, und dann zu einem späteren Zeitpunkt einen Sicherheits-Blueprint anwenden. Wenn einer der Blueprints aktualisiert wird, können Sie die Änderungen oder Korrekturen über das Lebenszyklusmanagement in Ihr Projekt integrieren. Weitere Informationen finden Sie unter [Referenz zum Projekt-Blueprint](#) und [Als Blueprint-Benutzer mit Lifecycle Management arbeiten](#).

Als Blueprint-Autor können Sie auch benutzerdefinierte Blueprints erstellen und veröffentlichen, damit Ihre CodeCatalyst Space-Mitglieder Ihre Projektressourcen nutzen können. Die benutzerdefinierten Blueprints können so entwickelt werden, dass sie den spezifischen Anforderungen für die Projekte Ihres Bereichs entsprechen. Nachdem Sie dem Blueprint-Katalog Ihres Bereichs einen benutzerdefinierten Blueprint hinzugefügt haben, können Sie den Blueprint verwalten und weitere Aktualisierungen vornehmen, sodass die Projekte Ihres Bereichs stets auf dem neuesten Stand der bewährten Methoden sind. Weitere Informationen finden Sie unter [Arbeiten mit benutzerdefinierten Blueprints in CodeCatalyst](#). [Das Blueprints-SDK und Beispiel-Blueprints finden Sie im Open-Source-Repository. GitHub](#)

Themen

- [Ein Projekt mit einem Blueprint erstellen](#)
- [Blueprints in einem Projekt anwenden und deren Zuordnung aufheben](#)
- [Aktualisierung eines Blueprints in einem Projekt](#)
- [Bearbeiten einer Beschreibung für einen Blueprint in einem Projekt](#)
- [Als Blueprint-Benutzer mit Lifecycle Management arbeiten](#)
- [Referenz zum Projekt-Blueprint](#)
- [Arbeiten mit benutzerdefinierten Blueprints in CodeCatalyst](#)
- [Kontingente für Blueprints in CodeCatalyst](#)

## Ein Projekt mit einem Blueprint erstellen

Sie können schnell ein Projekt mit einem Blueprint aus dem CodeCatalyst Amazon-Katalog oder dem Space-Katalog Ihres Teams mit benutzerdefinierten Blueprints erstellen. Je nach Blueprint wird Ihr Projekt mit bestimmten Ressourcen erstellt. Weitere Informationen finden Sie unter [Ein Projekt mit einem Blueprint erstellen](#) und [Referenz zum Projekt-Blueprint](#).

Nachdem Sie ein Projekt erstellt haben, können Sie zusätzliche Blueprints aus dem CodeCatalyst Katalog oder dem Katalog Ihres Raums mit benutzerdefinierten Blueprints auf Ihr CodeCatalyst Projekt anwenden. Blueprints stellen architektonische Komponenten dar, sodass mehrere Blueprints zusammen in deinem Projekt verwendet werden können, um die Best Practices deines Teams zu integrieren. Auf diese Weise können Sie auch sicherstellen, dass Ihr Projekt über die neuesten Änderungen an den sich entwickelnden Komponenten auf dem neuesten Stand ist. Weitere Informationen zum Arbeiten mit Blueprints in Ihrem Projekt finden Sie unter [Als Blueprint-Benutzer mit Lifecycle Management arbeiten](#).

## Blueprints in einem Projekt anwenden und deren Zuordnung aufheben

Sie können mehrere Blueprints in einem Projekt anwenden, um funktionale Komponenten, Ressourcen und Steuerung zu integrieren. Ihre Projekte können verschiedene Elemente unterstützen, die unabhängig voneinander in separaten Blueprints verwaltet werden. Die Anwendung von Blueprints auf ein Projekt reduziert die Notwendigkeit, Ressourcen manuell zu erstellen und Softwarekomponenten funktionsfähig zu machen. Ihre Projekte können auch auf dem neuesten Stand bleiben, wenn sich die Anforderungen ändern. Wenn Ihr Projekt keine Ressourcen aus einem

Blueprint mehr benötigt, können Sie den Blueprint von Ihrem Projekt trennen. Weitere Informationen zum Anwenden von Blueprints in Ihrem Projekt finden Sie unter [Als Blueprint-Benutzer mit Lifecycle Management arbeiten](#)

## Themen

- [Einen Blueprint auf ein Projekt anwenden](#)
- [Aufheben der Zuordnung eines Blueprints zu einem Projekt](#)

## Einen Blueprint auf ein Projekt anwenden

Um einen Blueprint auf Ihr Projekt anzuwenden

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich und wählen Sie dann das Projekt aus, auf das Sie einen Blueprint anwenden möchten.
3. Wählen Sie im Navigationsbereich Blueprints und dann Blueprint anwenden aus.
4. Wählen Sie auf der Registerkarte CodeCatalyst Blueprints einen Blueprint oder auf der Registerkarte Space-Blueprints einen benutzerdefinierten Blueprint aus, und wählen Sie dann Weiter aus.
5. Wählen Sie unter Blueprint-Details eine Blueprint-Version aus dem Dropdownmenü Zielversion aus. Die neueste Version wird automatisch ausgewählt.
6. Konfigurieren Sie unter Blueprint konfigurieren die Blueprint-Parameter.
7. Überprüfen Sie die Unterschiede zwischen der aktuellen Blueprint-Version und Ihrer aktualisierten Version. Der in einer Pull-Anfrage angezeigte Unterschied zeigt die Änderungen zwischen der aktuellen Version und der neuesten Version, die zum Zeitpunkt der Erstellung der Pull-Anfrage die gewünschte Version ist. Wenn keine Änderungen angezeigt werden, sind die Versionen möglicherweise identisch, oder Sie haben dieselbe Version sowohl für die aktuelle Version als auch für die gewünschte Version ausgewählt.
8. Wenn Sie überzeugt sind, dass die Pull-Anfrage den Code und die Änderungen enthält, die Sie überprüfen möchten, wählen Sie Blueprint anwenden aus. Nachdem Sie den Pull Request erstellt haben, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @ Zeichen, gefolgt vom Namen der Datei, verwendest.

**Note**

Der Blueprint wird erst angewendet, wenn der Pull-Request genehmigt und zusammengeführt wurde. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

Blueprint-Autoren können einen benutzerdefinierten Blueprint auch auf Projekte in bestimmten Bereichen anwenden, für die der Blueprint nicht verfügbar ist, um neue Projekte zu erstellen oder auf bestehende Projekte anzuwenden. Weitere Informationen finden Sie unter [Veröffentlichen und Anwenden eines benutzerdefinierten Blueprints in bestimmten Bereichen und Projekten](#).

## Aufheben der Zuordnung eines Blueprints zu einem Projekt

Wenn Sie keine neuen Updates zu einem Blueprint wünschen, können Sie die Zuordnung des Blueprints zu Ihrem Projekt aufheben. Ressourcen und funktionale Softwarekomponenten, die Ihrem Projekt aus dem Blueprint hinzugefügt wurden, bleiben in Ihrem Projekt erhalten.

Um einen Blueprint von Ihrem Projekt zu trennen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich und wählen Sie dann das Projekt aus, zu dem Sie die Zuordnung zu einem Blueprint aufheben möchten.
3. Wählen Sie im Navigationsbereich die Option Blueprints aus.
4. Wählen Sie den Blueprint mit den Ressourcen aus, deren Zuordnung Sie aufheben möchten, wählen Sie das Dropdownmenü Aktionen und dann Blueprint trennen aus.
5. Geben Sie die Eingabetaste einconfirm, um die Trennung zu bestätigen.
6. Wählen Sie Bestätigen aus.


## Aktualisierung eines Blueprints in einem Projekt

Wenn Sie ein Projekt mit einem Blueprint erstellt oder einen Blueprint auf ein vorhandenes Projekt angewendet haben, werden Sie über neue Versionen des Blueprints informiert. Bevor die Blueprint-Version durch eine genehmigte Pull-Anfrage aktualisiert wird, können Sie sich die Codeänderungen und die betroffenen Umgebungen ansehen. Mit dem Lebenszyklusmanagement können Sie einen oder mehrere angewendete Blueprints in Ihrem Projekt aktualisieren, sodass jeder Blueprint

aktualisiert werden kann, ohne dass sich dies auf andere Bereiche Ihres Projekts auswirkt. Sie können Blueprint-Aktualisierungen auch überschreiben. Weitere Informationen finden Sie unter [Als Blueprint-Benutzer mit Lifecycle Management arbeiten](#).

Um einen Blueprint auf die neueste Version zu aktualisieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie die Version eines Blueprints aktualisieren möchten.
3. Wählen Sie im Space-Dashboard das Projekt mit dem Blueprint aus, den Sie aktualisieren möchten.
4. Wählen Sie im Navigationsbereich Blueprints und dann das Optionsfeld für den Blueprint aus, den Sie aktualisieren möchten.
5. Wählen Sie das Dropdownmenü Aktionen und dann Version aktualisieren aus.
6. Wählen Sie im Dropdownmenü Zielversion die Version aus, die Sie für Ihr Update verwenden möchten. Die neueste Version wird automatisch ausgewählt.
7. Konfigurieren Sie unter Blueprint konfigurieren die Blueprint-Parameter.
8. Überprüfen Sie die Unterschiede zwischen der aktuellen Blueprint-Version und der aktualisierten Version. Der in einer Pull-Anfrage angezeigte Unterschied besteht in den Änderungen zwischen der aktuellen Version und der neuesten Version, die zum Zeitpunkt der Erstellung der Pull-Anfrage die gewünschte Version ist. Wenn keine Änderungen angezeigt werden, sind die Versionen möglicherweise identisch, oder Sie haben dieselbe Version sowohl für die aktuelle Version als auch für die gewünschte Version ausgewählt.
9. Wenn Sie überzeugt sind, dass der Pull Request den Code und die Änderungen enthält, die Sie überprüfen möchten, wählen Sie Update anwenden aus. Nachdem du den Pull Request erstellt hast, kannst du Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien und zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @ Zeichen, gefolgt vom Namen der Datei, verwendest.

 Note

Der Blueprint wird erst aktualisiert, wenn der Pull-Request genehmigt und zusammengeführt wurde. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

**Note**

Wenn Sie bereits Pull-Requests für die Aktualisierung einer Blueprint-Version geöffnet haben, schließen Sie die vorherigen Pull-Requests, bevor Sie eine neue erstellen. Wenn Sie Version aktualisieren auswählen, werden Sie zur Liste der ausstehenden Pull-Requests für den Blueprint weitergeleitet. Sie können ausstehende Pull Requests auch auf der Registerkarte Blueprints in den Projekteinstellungen und auf der Seite mit der Projektzusammenfassung anzeigen. Weitere Informationen finden Sie unter [Pull-Requests anzeigen](#).

## Bearbeiten einer Beschreibung für einen Blueprint in einem Projekt

Sie können die Beschreibung eines Blueprints bearbeiten, den Sie zur Erstellung eines Projekts verwendet oder den Sie nach der Erstellung eines Projekts angewendet haben. Ein Blueprint kann in einem Projekt mehr als einmal verwendet werden. Um den Zweck von Blueprints in Ihrem Projekt zu unterscheiden, können Sie Beschreibungen für diese Blueprints verwenden. Beschreibungen können auch verwendet werden, um die Komponenten, die Sie anwenden, anhand eines bestimmten Blueprints zu identifizieren.

Um die Beschreibung eines Blueprints in Ihrem Projekt zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu Ihrem Bereich und wählen Sie dann das Projekt mit den Blueprint-Einstellungen aus, das Sie aktualisieren möchten.
3. Wählen Sie im Navigationsbereich die Option Blueprints aus.
4. Wählen Sie den Blueprint mit der Beschreibung aus, den Sie aktualisieren möchten, wählen Sie das Dropdownmenü Aktionen und dann Beschreibung bearbeiten aus.
5. Geben Sie im Texteingabefeld für die Blueprint-Beschreibung eine Beschreibung ein, um den Blueprint in Ihrem Projekt zu identifizieren.
6. Klicken Sie auf Speichern.

# Als Blueprint-Benutzer mit Lifecycle Management arbeiten

Lebenszyklusmanagement ist die Fähigkeit, eine Codebasis aus aktualisierten Optionen oder Versionen eines Blueprints neu zu generieren. Auf diese Weise kann ein Blueprint-Autor den Softwareentwicklungszyklus jedes Projekts, das einen bestimmten Blueprint enthält, zentral verwalten. Wenn beispielsweise ein Sicherheitsupdate auf einen Webanwendungs-Blueprint übertragen wird, kann jedes Projekt, das den Webanwendungs-Blueprint enthält oder anhand dessen erstellt wurde, diesen Fix automatisch übernehmen. Dasselbe Management-Framework ermöglicht es Ihnen als Blueprint-Benutzer auch, Blueprint-Optionen zu ändern, nachdem sie ausgewählt wurden.

## Themen

- [Verwenden des Lebenszyklusmanagements für bestehende Projekte](#)
- [Verwenden des Lebenszyklusmanagements für mehrere Blueprints in einem Projekt](#)
- [Umgang mit Konflikten in Lifecycle-Pull-Requests](#)
- [Abmeldung von Änderungen im Lebenszyklusmanagement](#)
- [Überschreiben des Lebenszyklusmanagements eines Blueprints in einem Projekt](#)

## Verwenden des Lebenszyklusmanagements für bestehende Projekte

Sie können das Lebenszyklusmanagement für Projekte verwenden, die anhand von Blueprints erstellt wurden, oder für bestehende Projekte, die keinen Blueprints zugeordnet sind. Sie können beispielsweise einer five-year-old Java-Anwendung, die nie anhand eines Blueprints erstellt wurde, einen Standard-Blueprint für Sicherheitspraktiken hinzufügen. Der Blueprint generiert einen Workflow für Sicherheitsscans und anderen zugehörigen Code. Dieser Teil der Codebasis in der Java-Anwendung wird nun bei jeder Änderung des Blueprints automatisch mit den Best Practices Ihres Teams auf dem neuesten Stand gehalten.

## Verwenden des Lebenszyklusmanagements für mehrere Blueprints in einem Projekt

Da Blueprints architektonische Komponenten darstellen, können mehrere Blueprints häufig zusammen in demselben Projekt verwendet werden. Ein Projekt könnte beispielsweise aus einem zentralen Web-API-Blueprint bestehen, der von einem Plattformentwickler des Unternehmens erstellt wurde, sowie einem vom App-Sicherheitsteam erstellten Blueprint für die Versionsprüfung. Jeder dieser Blueprints kann unabhängig aktualisiert werden und erinnert sich an die in der Vergangenheit für ihn geltenden Zusammenführungslösungen.



**Note**

Da es sich um beliebige Architekturkomponenten handelt, sind nicht alle Blueprints zusammen sinnvoll oder funktionieren logisch zusammen, obwohl sie immer noch versuchen, miteinander zu verschmelzen.

## Umgang mit Konflikten in Lifecycle-Pull-Requests

Gelegentlich können Lifecycle-Pull-Anfragen zu Merge-Konflikten führen. Diese können manuell gelöst werden. Auflösungen werden bei nachfolgenden Blueprint-Updates gespeichert.

## Abmeldung von Änderungen im Lebenszyklusmanagement

Benutzer können einen Blueprint aus einem Projekt entfernen, um alle Verweise auf den Blueprint aufzuheben und Lebenszyklus-Updates zu deaktivieren. Aus Sicherheitsgründen werden dadurch weder der Code noch die Ressourcen des Projekts entfernt oder beeinträchtigt, auch nicht das, was aus dem Blueprint hinzugefügt wurde. Weitere Informationen finden Sie unter [Aufheben der Zuordnung eines Blueprints zu einem Projekt](#).

## Überschreiben des Lebenszyklusmanagements eines Blueprints in einem Projekt

Wenn Sie die Aktualisierungen eines Blueprints an bestimmten Dateien in Ihrem Projekt überschreiben möchten, können Sie eine Eigentümerdatei in Ihr Repository aufnehmen. GitLabDie Spezifikation [von Code Owners](#) entspricht den empfohlenen Richtlinien. Der Blueprint berücksichtigt immer die Datei mit den Code-Besitzern vor allem anderen und kann ein Beispiel wie das Folgende generieren:

```
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/**',
```

```
    ],  
  },  
],  
},  
});
```

Dadurch wird eine `.ownership`-file mit dem folgenden Inhalt generiert:

```
[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/**
```

## Referenz zum Projekt-Blueprint

Wenn Sie ein Projekt mithilfe eines Blueprints erstellen, CodeCatalyst wird ein vollständiges Projekt mit einem Quell-Repository, Beispielquellcode, CI/CD-Workflows, Build- und Testberichten sowie integrierten Tools zur Problemverfolgung erstellt. Ein Projekt-Blueprint verwendet Code, um Cloud-Infrastruktur, Ressourcen und Beispielquellenartefakte für verschiedene Arten von Anwendungen und Frameworks bereitzustellen.

Weitere Informationen finden Sie unter [Ein Projekt in Amazon erstellen CodeCatalyst](#). Sie müssen der Space-Administrator sein, um ein Projekt erstellen zu können.

Themen

- [Verfügbare Blueprints](#)
- [Informationen zum Projekt-Blueprint finden](#)

## Verfügbare Blueprints

Name des Blueprints	Beschreibung des Blueprints
ASP.NET Core-Web-API	Mit diesem Blueprint wird eine ASP.NET Core-Web-API-Anwendung für .NET 6 erstellt. Der Blueprint verwendet das AWS Deployment-Tool für.NET und bietet eine Option zur Konfigura

Name des Blueprints	Beschreibung des Blueprints
	tion von Amazon Elastic Container Service oder AWS Elastic Beanstalk als Bereitstellungsziel. AWS App Runner
AWS Glue ETL	Dieser Blueprint erstellt eine Beispielimplementierung für Extract Transform Load (ETL), die AWS CDK, AWS Glue, AWS Lambda und Amazon Athena verwendet, um kommagetrennte Werte (CSVs) in Apache Parquet zu konvertieren.
DevOps Bereitstellungspipeline	Mit diesem Blueprint wird mithilfe der Deployment Pipeline-Referenzarchitektur eine AWS Bereitstellungspipeline erstellt, in der eine Referenzanwendung AWS über mehrere Phasen hinweg bereitgestellt wird.
Java-API mit AWS Fargate	Dieser Blueprint erstellt ein containerisiertes Webservice-Projekt. Das Projekt verwendet <a href="#">AWS Copilot CLI</a> , um einen containerisierten <a href="#">Spring Boot</a> Java-Webservice zu erstellen und bereitzustellen, der von Amazon DynamoDB auf Amazon ECS unterstützt wird. Das Projekt stellt eine containerisierte App auf einem Amazon ECS-Cluster auf AWS Fargate serverlosem Computing bereit. Die App speichert Daten in einer DynamoDB-Tabelle. Nachdem Ihr Workflow erfolgreich ausgeführt wurde, ist der Beispielwebdienst über den Application Load Balancer öffentlich verfügbar.
Moderne dreistufige Webanwendung	Dieser Blueprint generiert Code in Python für die Anwendungsebene und das Vue-Frontend-Framework, um eine gut strukturierte, moderne 3-Tier-Webanwendung zu erstellen und bereitzustellen.

Name des Blueprints	Beschreibung des Blueprints
.NET-Anwendung ohne Server	Dieser Blueprint erstellt AWS Lambda Funktionen mithilfe von .NET CLI Lambda-Tools. Der Blueprint bietet Optionen für die AWS Lambda Funktionen, darunter die Wahl zwischen C# oder F#.
Node.js API mit AWS Fargate	Dieser Blueprint erstellt ein containerisiertes Webservice-Projekt. Das Projekt verwendet <a href="#">AWS Copilot CLI</a> , um einen containerisierten <a href="#">Express/Node.js</a> Webservice auf Amazon Elastic Container Service zu erstellen und bereitzustellen. Das Projekt stellt eine containerisierte App auf einem Amazon ECS-Cluster auf AWS Fargate serverlosem Computing bereit. Nachdem Ihr Workflow erfolgreich ausgeführt wurde, ist der Beispielwebdienst über den Application Load Balancer öffentlich verfügbar.
Serverloses Anwendungsmodell (SAM)	Mit diesem Blueprint wird ein Projekt erstellt, das ein serverloses Anwendungsmodell (SAM) verwendet, um eine API zu erstellen und bereitzustellen. Sie können SDK for Java oder SDK für Python als Programmiersprache wählen. TypeScript
Serverloser Image-Handler	Dieser Blueprint erstellt eine Anwendung für Hochgeschwindigkeits-Bildverarbeitung, ohne die Bildqualität zu beeinträchtigen.
Serverloser RESTful-Mikroservice	Dieser Blueprint erstellt eine REST-API, die eine To-Do-Dienstreferenz verwendet AWS Lambda und Amazon API Gateway mit dieser verknüpft ist. Sie können SDK for Java oder SDK für Python als Programmiersprache wählen. TypeScript

Name des Blueprints	Beschreibung des Blueprints
Einseitige Anwendung	Dieser Blueprint erstellt eine einseitige Anwendung (SPA), die React-, Vue- und Angular-Frameworks verwendet. Wählen Sie für das AWS Amplify Hosting zwischen Hosting oder Amazon CloudFront Amazon S3.
Statische Website	Dieser Blueprint erstellt eine statische Website mit den Generatoren für statische Websites von <a href="#">Hugo</a> oder <a href="#">Jekyll</a> . Generatoren für statische Websites verwenden Texteingabedateien (wie Markdown), um statische Webseiten zu generieren. Sie eignen sich ideal für selten wechselnde, informative Inhalte wie Produktseiten, Dokumentationen und Blogs. Der Blueprint verwendet die AWS CDK , um statische Webseiten AWS Amplify entweder auf Amazon S3 CloudFront S3+bereitzustellen.
To Do-Webanwendung	Dieser Blueprint erstellt eine serverlose To Do-Webanwendung mit Frontend- und Backend-Komponenten. Sie können SDK for Java oder SDK für Python als Programmiersprache wählen. TypeScript
ideo-on-demand V-Webdienst	Mit diesem Blueprint wird ein video-on-demand Dienst erstellt, der die Möglichkeit bietet, Inhalte aufzunehmen, zu transkodieren und bereitzustellen. Der Blueprint verwendet AWS Lambda Amazon S3, Amazon CloudWatch, und AWS Elemental MediaConvert.

Name des Blueprints	Beschreibung des Blueprints
Abonnieren Sie den externen Blueprint	<p>Dieser Blueprint erstellt einen Workflow für jedes importierte Paket. Diese Workflows werden einmal täglich ausgeführt, um NPM auf neue Versionen der Pakete zu überprüfen. Wenn eine neue Version vorhanden ist, versucht der Workflow, sie als benutzerdefinierten Blueprint zu Ihrem CodeCatalyst-Bereich hinzuzufügen. Die Aktion schlägt fehl, wenn ein Paket nicht gefunden werden kann oder es sich nicht um einen Blueprint handelt. Das Zielpaket muss sich auf NPM befinden und das Paket muss ein Blueprint sein. Der Space muss auf einer Stufe abonniert werden, die benutzerdefinierte Blueprints unterstützt.</p>
Bedrock GenAI-Chatbot	<p>Dieser Entwurf erstellt einen generativen KI-Chatbot mit <a href="#">Amazon Bedrock</a> und Claude von <a href="#">Anthropic</a>. Mit diesem Blueprint können Sie Ihren eigenen sicheren, durch Anmeldung geschützten LLM-Spielplatz erstellen und bereitstellen, der an Ihre Daten angepasst werden kann. Weitere Informationen finden Sie in der <a href="#">Bedrock GenAI Chatbot-Dokumentation</a>.</p>

Name des Blueprints	Beschreibung des Blueprints
Pläne für das AWS Project Development Kit (AWS PDK)	Diese PDK-Blueprints können zusammengesetzt werden, um eine Anwendung zu erstellen, die aus einer React-Website, einer Smithy-API und der unterstützenden CDK-Infrastruktur für die Bereitstellung in AWS besteht. Das AWS PDK bietet Bausteine für gängige Muster sowie Entwicklungstools zur Verwaltung und Erstellung Ihrer Projekte. Weitere Informationen finden Sie im <a href="#">AWS GitHub PDK-Quell-Repository</a> und <a href="#">Tutorial: Erstellen einer Full-Stack-Anwendung mit zusammensetzbaren PDK-Blueprints</a> .

## Informationen zum Projekt-Blueprint finden

Es sind mehrere Projekt-Blueprints verfügbar. CodeCatalyst Zu jedem Blueprint gibt es eine zugehörige Zusammenfassung und eine README-Datei. In der Zusammenfassung werden die Ressourcen beschrieben, die durch den Blueprint installiert werden, während die README-Datei den Blueprint ausführlich erklärt und Anweisungen zu seiner Verwendung enthält.

## Arbeiten mit benutzerdefinierten Blueprints in CodeCatalyst

Sie können die Entwicklung und die Best Practices für die Projekte Ihres CodeCatalyst Bereichs mit benutzerdefinierten Blueprints standardisieren. Benutzerdefinierte Blueprints können verwendet werden, um verschiedene Aspekte eines CodeCatalyst Projekts zu definieren, z. B. Workflow-Definitionen und Anwendungscode. Nachdem ein benutzerdefinierter Blueprint verwendet wurde, um ein neues Projekt zu erstellen oder auf bestehende Projekte angewendet zu haben, sind alle Änderungen am Blueprint für diese Projekte als Pull-Request-Updates verfügbar. Als Blueprint-Autor können Sie Details darüber einsehen, welche Projekte Ihre Blueprints in Ihrem gesamten Bereich verwenden, sodass Sie sehen können, wie Standards projektübergreifend angewendet werden. Das Lebenszyklusmanagement eines Blueprints ermöglicht es Ihnen, den Softwareentwicklungszyklus jedes Projekts zentral zu verwalten. So können Sie sicherstellen, dass die Projekte in Ihrem Bereich weiterhin den Best Practices mit den neuesten Änderungen oder Korrekturen folgen. Weitere Informationen finden Sie unter [Arbeiten mit Lebenszyklusmanagement als Blueprint-Autor](#).

Benutzerdefinierte Blueprints bieten die Möglichkeit, Blueprint-Versionen durch Resynthese mit denen des vorherigen Projekts zu aktualisieren. Resynthese ist der Vorgang, bei dem die Blueprint-Synthese mit aktualisierten Versionen erneut ausgeführt wird oder die Möglichkeit besteht, Korrekturen und Änderungen in bestehende Projekte zu integrieren. Weitere Informationen finden Sie unter [Konzepte für benutzerdefinierte Vorlagen](#).

[Das Blueprints-SDK und Beispiel-Blueprints finden Sie im Open-Source-Repository. GitHub](#)

## Themen

- [Konzepte für benutzerdefinierte Vorlagen](#)
- [Erste Schritte mit benutzerdefinierten Blueprints](#)
- [Tutorial: Eine React-Anwendung erstellen und aktualisieren](#)
- [Arbeiten mit Lebenszyklusmanagement als Blueprint-Autor](#)
- [Entwickeln eines benutzerdefinierten Blueprints](#)
- [Veröffentlichen eines benutzerdefinierten Blueprints](#)
- [Details, Versionen und Projekte eines benutzerdefinierten Blueprints anzeigen](#)
- [Hinzufügen und Entfernen eines benutzerdefinierten Blueprints in einem Bereich](#)
- [Veröffentlichungsberechtigungen für einen benutzerdefinierten Blueprint verwalten](#)
- [Versionen für einen benutzerdefinierten Blueprint verwalten](#)
- [Löschen eines veröffentlichten benutzerdefinierten Blueprints oder einer veröffentlichten Version](#)
- [Arbeiten mit Abhängigkeiten und Tools](#)
- [Beitragend](#)

## Konzepte für benutzerdefinierte Vorlagen

Im Folgenden finden Sie einige Konzepte und Begriffe, die Sie kennen sollten, wenn Sie mit benutzerdefinierten Blueprints in arbeiten CodeCatalyst.

## Themen

- [Blueprint-Projekt](#)
- [Raumvorlagen](#)
- [Katalog für Raumvorlagen](#)
- [Synthesis](#)



- [Erneute Synchronisierung](#)
- [Teilweise Optionen](#)
- [Projen](#)

## Blueprint-Projekt

Ein Blueprint-Projekt bietet Ihnen die Möglichkeit, Blueprints in Ihrem Bereich zu entwickeln und zu veröffentlichen. Während der Projekterstellung wird ein Quell-Repository erstellt, und der Name des Repositories ist der, den Sie bei der Eingabe der Projektressourcendetails ausgewählt haben. Wenn Sie während der Blueprint-Erstellung eine Workflow-Version generieren möchten, wird in Ihrem Blueprint ein Veröffentlichungs-Workflow mit dem Blueprint Builder erstellt. Der Workflow veröffentlicht automatisch Ihre neueste Version.

## Raumvorlagen

Sie können alle Vorlagen in der Tabelle Raumvorlagen anzeigen und verwalten, wenn Sie zum Abschnitt Vorlagen Ihres Bereichs navigieren. Nachdem Blueprints in Ihrem Bereich veröffentlicht wurden, werden sie als Raumvorlage zur Verfügung gestellt, die dem Blueprint-Katalog Ihres Bereichs hinzugefügt und daraus entfernt werden soll. Sie können auch Veröffentlichungsberechtigungen verwalten und Blueprints aus im Bereich Blueprints Ihres Bereichs löschen. Weitere Informationen finden Sie unter [Details, Versionen und Projekte eines benutzerdefinierten Blueprints anzeigen](#).

## Katalog für Raumvorlagen

Sie können alle hinzugefügten benutzerdefinierten Blueprints aus dem Blueprint-Katalog eines Raums anzeigen. Hier kann ein Leerzeichen Ihren benutzerdefinierten Blueprint auswählen, um ein neues Projekt zu erstellen. Dieser Katalog unterscheidet sich vom CodeCatalyst Katalog, der bereits über verfügbare Vorlagen für alle Raummitglieder verfügt. Weitere Informationen finden Sie unter [Referenz zum Projekt-Blueprint](#).

## Synthesis

Synthesis ist der Prozess der Generierung eines CodeCatalyst Projektpakets, das den Quellcode, die Konfiguration und die Ressourcen in einem Projekt darstellt. Das Paket wird dann von CodeCatalyst Bereitstellungs-API-Operationen verwendet, um in einem Projekt bereitzustellen. Der Prozess kann lokal ausgeführt werden, während Sie Ihren benutzerdefinierten Blueprint entwickeln, um die

Projekterstellung zu emulieren, ohne ein Projekt in erstellen zu müssen CodeCatalyst. Die folgenden Befehle können verwendet werden, um die Synthetisierung durchzuführen:

```
yarn blueprint:synth          # fast mode
yarn blueprint:synth --cache  # wizard emulation mode
```

Der Blueprint beginnt mit dem Aufruf der `blueprint.ts` Hauptklasse mit dieser Option, die auf `wurdedefaults.json` zusammengeführt wurde. Ein neues Projektpaket wird unter dem `synth/synth.[options-name]/proposed-bundle/` Ordner generiert. Die Ausgabe enthält das Projektpaket, das ein benutzerdefinierter Blueprint generiert, unter Berücksichtigung der von Ihnen festgelegten Optionen, einschließlich der [teilweisen Optionen](#), die Sie möglicherweise konfiguriert haben.

## Erneute Synchronisierung

Erneute Synchronisierung ist der Prozess der Generierung eines Blueprints mit unterschiedlichen Blueprint-Optionen oder Blueprint-Versionen vorhandener Projekte. Als Blueprint-Autor können Sie benutzerdefinierte Zusammenführungsstrategien im benutzerdefinierten Blueprint-Code definieren. Sie können auch Eigentums Grenzen in einer `.ownership-file` definieren, um anzugeben, in welchen Teilen der Codebasis ein Blueprint aktualisiert werden darf. Während der benutzerdefinierte Blueprint Aktualisierungen für vorschlagen kann `.ownership-file`, können Projektentwickler, die den benutzerdefinierten Blueprint verwenden, die Eigentums Grenzen für ihre Projekte bestimmen. Sie können die Neusyntax lokal ausführen und testen und aktualisieren, bevor Sie Ihren benutzerdefinierten Blueprint veröffentlichen. Verwenden Sie die folgenden Befehle, um eine Neusynchronisierung durchzuführen:

```
yarn blueprint:resynth       # fast mode
yarn blueprint:resynth --cache # wizard emulation mode
```

Der Blueprint beginnt mit dem Aufruf der `blueprint.ts` Hauptklasse mit dieser Option, die auf `wurdedefaults.json` zusammengeführt wurde. Ein neues Projektpaket wird unter dem `synth/resynth.[options-name]/` Ordner generiert. Die Ausgabe enthält das Projektpaket, das ein benutzerdefinierter Blueprint generiert, unter Berücksichtigung der von Ihnen festgelegten Optionen, einschließlich der [teilweisen Optionen](#), die Sie möglicherweise konfiguriert haben.

Der folgende Inhalt wird nach den Synthetisierungs- und Resynthetikprozessen erstellt:

- Empfohlenes Paket – Die Ausgabe der -Synthetik, wenn sie mit neuen Optionen für die Ziel-Blueprint-Version ausgeführt wird.

- `existing-bundle` – Ein Mock Ihres vorhandenen Projekts. Wenn in diesem Ordner nichts vorhanden ist, wird es mit derselben Ausgabe wie `generiertproposed-bundle`.
- Vorgänger-Paket – Ein Mock dessen, was Ihr Blueprint generieren würde, wenn er entweder mit einer früheren Version, früheren Optionen oder einer Kombination ausgeführt wird. Wenn in diesem Ordner nichts vorhanden ist, wird es mit derselben Ausgabe wie `generiertproposed-bundle`.
- aufgelöstes Paket – Das Paket wird immer neu generiert und verwendet standardmäßig eine dreiseitige Zusammenführung zwischen `proposed-bundle`, `existing-bundle` und `ancestor-bundle`. Dieses Paket bietet eine Emulation dessen, was eine Resynthese lokal ausgeben würde.

Weitere Informationen zu Blueprint-Ausgabepaketten finden Sie unter [Generieren von Dateien mit Resynthese](#).

## Teilweise Optionen

Sie können Optionsvarianten unter `hinzufügensrc/wizard-configuration/`, die nicht die gesamte `Options` Schnittstelle aufzählen müssen, und die Optionen werden oben in der `defaults.json` Datei zusammengeführt. Auf diese Weise können Sie Testfälle an bestimmte Optionen anpassen.

Beispiel:

Options -Schnittstelle:

```
{
  language: "Python" | "Java" | "Typescript",
  repositoryName: string
  ...
}
```

`defaults.json`-Datei:

```
{
  language: "Python",
  repositoryName: "Myrepo"
  ...
}
```

Zusätzliche Konfigurationstests:

- ```
#wizard-config-typescript-test.json
{
  language: "Typescript",
}
```

- ```
#wizard-config-java-test.json
{
  language: "Java",
}
```

## Projen

Projen ist ein Open-Source-Tool, das benutzerdefinierte Blueprints verwenden, um sich auf dem neuesten Stand und konsistent zu halten. Blueprints werden als Projen-Pakete geliefert, da dieses Framework Ihnen die Möglichkeit bietet, Projekte zu erstellen, zu bündeln und zu veröffentlichen, und Sie die -Schnittstelle verwenden können, um die Konfigurationen und Einstellungen eines Projekts zu verwalten.

Sie können Projen verwenden, um Blueprints in großem Umfang zu aktualisieren, auch nachdem sie erstellt wurden. Das Projen-Tool ist die zugrunde liegende Technologie hinter der Blueprint-Synthetik, die ein Projektpaket generiert. Projen besitzt die Konfiguration für ein Projekt und sollte sich nicht auf Sie als Blueprint-Autor auswirken. Sie können ausführen `yaarn projen`, um die Konfiguration Ihres Projekts nach dem Hinzufügen von Abhängigkeiten neu zu generieren, oder Sie können Optionen in der `projenrc.ts` Datei ändern. Projen ist auch das zugrunde liegende Generationstool für benutzerdefinierte Blueprints zur Synthetisierung eines Projekts. Weitere Informationen finden Sie auf der [projen- GitHub Seite](#) . Weitere Informationen zur Arbeit mit Projen finden Sie in der [Projen-Dokumentation](#) und [So vereinfachen Sie die Projekteinrichtung mit Projen](#) .

## Erste Schritte mit benutzerdefinierten Blueprints

Während der Erstellung eines Blueprints können Sie den Blueprint konfigurieren und eine Vorschau der Projektressourcen generieren. Jeder benutzerdefinierte Blueprint wird von einem CodeCatalyst Projekt verwaltet, das standardmäßig einen Workflow für die Veröffentlichung im Blueprint-Katalog des Raums enthält.

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen eines benutzerdefinierten Blueprints in CodeCatalyst](#)

- [Schritt 2: Entwickeln eines benutzerdefinierten Blueprints mit Komponenten](#)
- [Schritt 3: Vorschau eines benutzerdefinierten Blueprints](#)
- [\(Optional\) Schritt 4: Veröffentlichen einer benutzerdefinierten Blueprint-Vorschauversion](#)

## Voraussetzungen

Berücksichtigen Sie vor dem Erstellen eines benutzerdefinierten Blueprints die folgenden Anforderungen:

- Ihr CodeCatalyst Bereich muss die Enterprise-Stufe sein. Weitere Informationen finden Sie unter [Verwalten der Fakturierung](#) im Amazon- CodeCatalyst Administratorhandbuch.
- Sie benötigen den Space-Administrator oder die Power-Benutzerrolle, um benutzerdefinierte Blueprints zu erstellen. Weitere Informationen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

## Schritt 1: Erstellen eines benutzerdefinierten Blueprints in CodeCatalyst

Wenn Sie einen benutzerdefinierten Blueprint aus den Einstellungen Ihres Bereichs erstellen, wird ein Repository für Sie erstellt. Das Repository enthält alle erforderlichen Ressourcen, die Sie benötigen, um Ihren Blueprint zu entwickeln, bevor Sie ihn im Blueprint-Katalog des Raums veröffentlichen.

So erstellen Sie einen benutzerdefinierten Blueprint

- a. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
- b. Navigieren Sie in der CodeCatalyst -Konsole zu dem Bereich, in dem Sie einen benutzerdefinierten Blueprint erstellen möchten.
- c. Wählen Sie im Leerzeichen-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
- d. Wählen Sie Blueprint erstellen aus.
- e. Geben Sie unter Benennen Ihres Blueprints den Namen ein, den Sie Ihrem Projekt zuweisen möchten, und die zugehörigen Ressourcennamen. Der Name muss in Ihrem Bereich eindeutig sein.
- f. Gehen Sie unter Blueprint-Details wie folgt vor:
  - i. Geben Sie im Textfeld Blueprint-Anzeigenname einen Namen ein, der im Blueprint-Katalog Ihres Raums angezeigt wird.

- ii. Geben Sie im Textfeld Beschreibung eine Beschreibung für Ihren benutzerdefinierten Blueprint ein.
- iii. Geben Sie im Textfeld Name des Autors einen Namen für Ihren benutzerdefinierten Blueprint ein.
- iv. (Optional) Wählen Sie die Option Erweiterte Einstellungen aus.
  - A. Wählen Sie + Hinzufügen, um der `package.json` Datei hinzugefügte Tags hinzuzufügen.
  - B. Wählen Sie das Dropdown-Menü Lizenz und dann eine Lizenz für Ihren benutzerdefinierten Blueprint aus.
  - C. Geben Sie in das Textfeld Blueprint-Paakge-Name einen Namen ein, um Ihr Blueprint-Paket zu identifizieren.
  - D. Standardmäßig wird ein Release-Workflow mit einem Veröffentlichungs-Blueprint in Ihrem Projekt namens Blueprint Builder generiert. Der Workflow veröffentlicht die neueste Blueprint-Version in Ihrem Bereich, wenn Sie Änderungen übertragen, da die Veröffentlichungsberechtigungen durch den Release-Workflow aktiviert sind. Um die Workflow-Generierung zu deaktivieren, deaktivieren Sie das Kontrollkästchen Workflow freigeben.
- g. (Optional) Ein Blueprint-Projekt verfügt über vordefinierten Code, um die Veröffentlichung des Blueprints im Raumkatalog zu unterstützen. Um Definitionsdateien mit Aktualisierungen anzuzeigen, die auf den von Ihnen vorgenommenen Projektparameterauswahlen basieren, wählen Sie Code anzeigen oder Workflow anzeigen unter Blueprint-Vorschau generieren aus.
- h. Wählen Sie Blueprint erstellen aus.

Wenn Sie die Workflow-Generierung für Ihren benutzerdefinierten Blueprint nicht deaktiviert haben, wird der Workflow automatisch ausgeführt, sobald Ihr Blueprint erstellt wird. Wenn die Workflow-Ausführung abgeschlossen ist, kann Ihr benutzerdefinierter Blueprint standardmäßig zum Blueprint-Katalog Ihres Raums hinzugefügt werden. Sie können die Veröffentlichungsberechtigungen deaktivieren, wenn Sie nicht möchten, dass die neueste Blueprint-Version automatisch in Ihrem Bereich veröffentlicht wird. Weitere Informationen finden Sie unter [Blueprints in einem Projekt anwenden und deren Zuordnung aufheben](#).

Da der Veröffentlichungs-Workflow mit dem Namen mit einem Blueprint erstellt `blueprint-release` wird, kann der Blueprint als angewendeter Blueprint in Ihrem Projekt gefunden werden.

Weitere Informationen finden Sie unter [Blueprints in einem Projekt anwenden und deren Zuordnung aufheben](#) und [Mit Workflows arbeiten](#).

## Schritt 2: Entwickeln eines benutzerdefinierten Blueprints mit Komponenten

Ein Blueprint-Assistent wird generiert, wenn Sie einen benutzerdefinierten Blueprint erstellen, und er kann bei der Entwicklung des benutzerdefinierten Blueprints mit Komponenten geändert werden. Sie können die `src/defaults.json` Dateien `src/blueprints.js` und aktualisieren, um den Assistenten zu ändern.

### Important

Wenn Sie Blueprint-Pakete aus externen Quellen verwenden möchten, sollten Sie die Risiken berücksichtigen, die mit diesen Paketen verbunden sein können. Sie sind für die benutzerdefinierten Vorlagen verantwortlich, die Sie Ihrem Bereich hinzufügen, und für den Code, den sie generieren.

Erstellen Sie eine Entwicklungsumgebung in Ihrem CodeCatalyst Projekt mit einer unterstützten integrierten Entwicklungsumgebung (IDE), bevor Sie Ihren Blueprint-Code konfigurieren. Eine Entwicklungsumgebung ist erforderlich, um mit den erforderlichen Tools und Paketen zu arbeiten.

So erstellen Sie eine Entwicklungsumgebung

1. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:
  - a. Wählen Sie Übersicht und navigieren Sie dann zum Abschnitt Meine Entwicklungsumgebungen.
  - b. Wählen Sie Code und dann Entwicklungsumgebungen aus.
  - c. Wählen Sie Code , Quell-Repositorys und wählen Sie das Repository aus, das Sie beim Erstellen Ihres Blueprints erstellt haben.
2. Wählen Sie Entwicklungsumgebung erstellen aus.
3. Wählen Sie eine unterstützte IDE aus dem Dropdown-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
4. Wählen Sie In vorhandener Verzweigung arbeiten und wählen Sie im Dropdown-Menü Vorhandene Verzweigung den Feature-Zweig aus, den Sie erstellt haben.
5. (Optional) Geben Sie im Feld Alias – optional eTexteingabe einen Alias ein, um die Entwicklungsumgebung zu identifizieren.

6. Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, zeigt die Spalte Status der Entwicklungsumgebung Start an und die Statusspalte zeigt Wird ausgeführt an, wenn die Entwicklungsumgebung erstellt wurde.

Weitere Informationen finden Sie unter [Entwicklungsumgebungen in CodeCatalyst](#).

So entwickeln Sie Ihren benutzerdefinierten Blueprint

1. Verwenden Sie in einem funktionierenden Terminal den folgenden yarn Befehl, um Abhängigkeiten zu installieren:

```
yarn
```

Die erforderlichen Tools und Pakete werden über die CodeCatalyst Entwicklungsumgebung zur Verfügung gestellt, einschließlich Yarn. Wenn Sie an einem benutzerdefinierten Blueprint ohne Entwicklungsumgebung arbeiten, installieren Sie zunächst Yarn auf Ihrem System. Weitere Informationen finden Sie in [der Yarn-Installationsdokumentation](#).

2. Entwickeln Sie Ihren benutzerdefinierten Blueprint so, dass er Ihren Einstellungen entspricht. Sie können den Assistenten Ihres Blueprints ändern, indem Sie Komponenten hinzufügen. Weitere Informationen finden Sie unter [Entwickeln eines benutzerdefinierten Blueprints](#), [Arbeiten mit einem Assistenten](#) und [Veröffentlichen eines benutzerdefinierten Blueprints](#).

### Schritt 3: Vorschau eines benutzerdefinierten Blueprints

Nachdem Sie Ihren benutzerdefinierten Blueprint eingerichtet und entwickelt haben, können Sie eine Vorschau anzeigen und die Vorschauversion Ihres Blueprints in Ihrem Bereich veröffentlichen. Eine Vorschauversion gibt Ihnen die Möglichkeit, zu überprüfen, ob der Blueprint Ihren Wünschen entspricht, bevor er zum Erstellen neuer Projekte verwendet oder auf bestehende Projekte angewendet wird.

So zeigen Sie eine Vorschau eines benutzerdefinierten Blueprints an

1. Verwenden Sie in einem funktionierenden Terminal den folgenden yarn Befehl:

```
yarn blueprint:preview
```

2. Navigieren Sie zu dem bereitgestellten `See this blueprint at:` Link, um eine Vorschau Ihrer benutzerdefinierten Vorlage anzuzeigen.



- Überprüfen Sie, ob die Benutzeroberfläche, einschließlich Text, je nach Konfiguration wie erwartet angezeigt wird. Wenn Sie Ihren benutzerdefinierten Blueprint ändern möchten, können Sie die `blueprint.ts` Datei bearbeiten, den Blueprint neu synchronisieren und dann erneut eine Vorschauversion veröffentlichen. Weitere Informationen finden Sie unter [Erneute Synchronisierung](#).

## (Optional) Schritt 4: Veröffentlichen einer benutzerdefinierten Blueprint-Vorschauversion

Sie können eine Vorschauversion Ihres benutzerdefinierten Blueprints in Ihrem Bereich veröffentlichen, wenn Sie sie dem Blueprints-Katalog Ihres Bereichs hinzufügen möchten. Auf diese Weise können Sie den Blueprint als Benutzer anzeigen, bevor Sie dem Katalog die Nicht-Vorschauversion hinzufügen. Mit der Vorschauversion können Sie veröffentlichen, ohne eine tatsächliche Version zu benötigen. Wenn Sie beispielsweise an einer `0.0.1` Version arbeiten, können Sie eine Vorschauversion veröffentlichen und hinzufügen, sodass neue Updates für eine zweite Version veröffentlicht und als hinzugefügt werden können `0.0.2`.

So veröffentlichen Sie eine Vorschauversion eines benutzerdefinierten Blueprints

Navigieren Sie zu dem bereitgestellten `Enable version [version number] at:` Link, um Ihren benutzerdefinierten Blueprint zu aktivieren. Dieser Link wird bereitgestellt, wenn der `yarn` Befehl in ausgeführt wird [Schritt 3: Vorschau eines benutzerdefinierten Blueprints](#).

Nachdem Sie Ihren benutzerdefinierten Blueprint erstellt, entwickelt, in der Vorschau angezeigt und veröffentlicht haben, können Sie die endgültige Blueprint-Version veröffentlichen und dem Blueprint-Katalog Ihres Raums hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen eines benutzerdefinierten Blueprints in einem Bereich](#).

## Tutorial: Eine React-Anwendung erstellen und aktualisieren

Als Autor von Blueprints können Sie benutzerdefinierte Blueprints entwickeln und dem Blueprints-Katalog Ihres Bereichs hinzufügen. Diese Blueprints können dann von Space-Mitgliedern verwendet werden, um neue Projekte zu erstellen oder sie auf bestehende Projekte anzuwenden. Sie können weiterhin Änderungen an Ihren Blueprints vornehmen, die dann als Updates über Pull-Requests zur Verfügung gestellt werden.

Dieses Tutorial bietet eine exemplarische Vorgehensweise aus der Sicht eines Blueprint-Autors und eines Blueprint-Benutzers. Das Tutorial zeigt, wie Sie einen einseitigen React-Webanwendungs-

Blueprint erstellen. Der Blueprint wird dann verwendet, um ein neues Projekt zu erstellen. Wenn der Blueprint mit Änderungen aktualisiert wird, werden diese Änderungen in das anhand des Blueprints erstellte Projekt über eine Pull-Anfrage übernommen.

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie einen benutzerdefinierten Blueprint](#)
- [Schritt 2: Freigabe-Workflow anzeigen](#)
- [Schritt 3: Blueprint zum Katalog hinzufügen](#)
- [Schritt 4: Projekt mit Blueprint erstellen](#)
- [Schritt 5: Blueprint aktualisieren](#)
- [Schritt 6: Aktualisieren Sie die veröffentlichte Katalogversion des Blueprints auf die neue Version](#)
- [Schritt 7: Aktualisieren Sie das Projekt mit der neuen Blueprint-Version](#)
- [Schritt 8: Sehen Sie sich die Änderungen im Projekt an](#)

## Voraussetzungen

Um einen benutzerdefinierten Blueprint zu erstellen und zu aktualisieren, müssen Sie die folgenden Aufgaben abgeschlossen haben: [Einrichten CodeCatalyst](#)

- Sie müssen über eine AWS Builder-ID verfügen, bei der Sie sich anmelden können CodeCatalyst.
- Gehören Sie einem Bereich an und lassen Sie sich in diesem Bereich die Rolle Space-Administrator oder Poweruser zuweisen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#), [Verwaltung von Space-Benutzern](#) und [Rolle „Space-Administrator“](#).

## Schritt 1: Erstellen Sie einen benutzerdefinierten Blueprint

Wenn Sie einen benutzerdefinierten Blueprint erstellen, wird ein CodeCatalyst Projekt erstellt, das Ihren Blueprint-Quellcode sowie Ihre Entwicklungstools und Ressourcen enthält. In Ihrem Projekt werden Sie den Blueprint entwickeln, testen und veröffentlichen.

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie einen Blueprint erstellen möchten.

3. Wählen Sie Einstellungen, um zu den Space-Einstellungen zu navigieren.
4. Wählen Sie auf der Registerkarte Space-Einstellungen die Option Blueprints und dann Blueprint erstellen aus.
5. Aktualisieren Sie die Felder im Assistenten zum Erstellen von Blueprints mit den folgenden Werten:
  - Geben Sie im Feld Blueprint-Name den Wert ein. `react-app-blueprint`
  - Geben Sie im Feld Blueprint-Anzeigename den Wert ein. `react-app-blueprint`
6. Wählen Sie optional Code anzeigen aus, um eine Vorschau des Blueprint-Quellcodes für Ihren Blueprint anzuzeigen. Wählen Sie ebenfalls Workflow anzeigen aus, um eine Vorschau des Workflows anzuzeigen, der in dem Projekt erstellt und veröffentlicht wird, das den Blueprint erstellt und veröffentlicht.
7. Wählen Sie Blueprint erstellen aus.
8. Sobald Ihr Blueprint erstellt wurde, werden Sie zum Projekt des Blueprints weitergeleitet. Dieses Projekt enthält den Blueprint-Quellcode sowie die Tools und Ressourcen, die Sie zum Entwickeln, Testen und Veröffentlichen des Blueprints benötigen. Es wurde ein Release-Workflow generiert, mit dem Ihr Blueprint automatisch im Space veröffentlicht wurde.
9. Nachdem Ihr Blueprint und Ihr Blueprint-Projekt erstellt wurden, besteht der nächste Schritt darin, es zu konfigurieren, indem Sie den Quellcode aktualisieren. Sie können Dev Environments verwenden, um Ihr Quell-Repository direkt in Ihrem Browser zu öffnen und zu bearbeiten.

Wählen Sie im Navigationsbereich Code und dann Dev Environments aus.

10. Wählen Sie Create Dev Environment und dann AWS Cloud9 (im Browser).
11. Behalten Sie die Standardeinstellungen bei und wählen Sie Create.
12. Navigieren Sie im AWS Cloud9 Terminal zu Ihrem Blueprint-Projektverzeichnis, indem Sie den folgenden Befehl ausführen:

```
cd react-app-blueprint
```

13. Ein `static-assets` Ordner wird automatisch erstellt und gefüllt, wenn ein Blueprint erstellt wird. In diesem Tutorial löschen Sie den Standardordner und generieren einen neuen Ordner für einen React-App-Blueprint.

Löschen Sie den Ordner `static-assets`, indem Sie den folgenden Befehl ausführen:

```
rm -r static-assets
```

AWS Cloud9 ist auf einer Linux-basierten Plattform aufgebaut. Wenn Sie ein Windows-Betriebssystem verwenden, können Sie stattdessen den folgenden Befehl verwenden:

```
rmdir /s /q static-assets
```

14. Nachdem der Standardordner gelöscht wurde, erstellen Sie einen `static-assets` Ordner für einen React-App-Blueprint, indem Sie den folgenden Befehl ausführen:

```
npx create-react-app static-assets
```

Wenn Sie dazu aufgefordert werden, geben Sie `ein`, um fortzufahren.

Eine neue React-Anwendung wurde in dem `static-assets` Ordner mit den erforderlichen Paketen erstellt. Die Änderungen müssen in Ihr CodeCatalyst Remote-Quell-Repository übertragen werden.

15. Stellen Sie sicher, dass Sie über die neuesten Änderungen verfügen, und übertragen Sie die Änderungen anschließend in das CodeCatalyst Quell-Repository des Blueprints, indem Sie die folgenden Befehle ausführen:

```
git pull
```

```
git add .
```

```
git commit -m "Add React app to static-assets"
```

```
git push
```

Wenn eine Änderung in das Quell-Repository des Blueprints übertragen wird, wird der Release-Workflow automatisch gestartet. Dieser Workflow erhöht die Blueprint-Version, erstellt den Blueprint und veröffentlicht ihn in Ihrem Bereich. Im nächsten Schritt navigieren Sie zum ausgeführten Release-Workflow, um zu sehen, wie er funktioniert.

## Schritt 2: Freigabe-Workflow anzeigen

1. Wählen Sie in der CodeCatalyst Konsole im Navigationsbereich CI/CD und dann Workflows aus.

2. Wählen Sie den Blueprint-Release-Workflow aus.
3. Sie können sehen, dass der Workflow Aktionen zum Erstellen und Veröffentlichen des Blueprints enthält.
4. Wählen Sie unter Letzte Ausführung den Link Workflow-Ausführung aus, um die Ausführung anhand der von Ihnen vorgenommenen Codeänderung anzuzeigen.
5. Sobald der Lauf abgeschlossen ist, wird Ihre neue Blueprint-Version veröffentlicht. Veröffentlichte Blueprint-Versionen sind in Ihren Space-Einstellungen sichtbar, können aber erst in Projekten verwendet werden, wenn sie dem Blueprints-Katalog des Spaces hinzugefügt wurden. Im nächsten Schritt fügen Sie den Blueprint dem Katalog hinzu.

### Schritt 3: Blueprint zum Katalog hinzufügen

Durch Hinzufügen eines Blueprints zum Blueprints-Katalog des Spaces kann der Blueprint in allen Projekten in einem Space verwendet werden. Space-Mitglieder können den Blueprint verwenden, um neue Projekte zu erstellen oder sie auf bestehende Projekte anzuwenden.

1. Navigieren Sie in der CodeCatalyst Konsole zurück zum Bereich.
2. Wählen Sie Einstellungen und dann Blueprints aus.
3. Wählen Sie react-app-blueprint und wählen Sie dann Zum Katalog hinzufügen aus.
4. Wählen Sie Speichern.

### Schritt 4: Projekt mit Blueprint erstellen

Nachdem der Blueprint dem Katalog hinzugefügt wurde, kann er in Projekten verwendet werden. In diesem Schritt erstellen Sie ein Projekt mit dem Blueprint, den Sie gerade erstellt haben. In einem späteren Schritt aktualisieren Sie dieses Projekt, indem Sie eine neue Version des Blueprints aktualisieren und veröffentlichen.


1. Wählen Sie die Registerkarte Projekte und dann Projekt erstellen aus.
2. Wählen Sie Space Blueprints und dann aus react-app-blueprint.

#### Note

Sobald der Blueprint ausgewählt ist, können Sie den Inhalt der Blueprint-Datei sehen.  
README.md

3. Wählen Sie Weiter aus.

4.

 Note

Der Inhalt dieses Assistenten zur Projekterstellung kann im Blueprint konfiguriert werden.

Geben Sie den Projektnamen als Blueprint-Benutzer ein. Geben Sie für dieses Tutorial `react-app-project` ein. Weitere Informationen finden Sie unter [Entwickeln eines benutzerdefinierten Blueprints](#).

Als Nächstes aktualisieren Sie den Blueprint und fügen die neue Version dem Katalog hinzu, mit dem Sie dieses Projekt aktualisieren werden.

## Schritt 5: Blueprint aktualisieren

Nachdem ein Blueprint zum Erstellen eines neuen Projekts verwendet oder auf bestehende Projekte angewendet wurde, können Sie als Blueprint-Autor weiterhin Aktualisierungen vornehmen. In diesem Schritt nehmen Sie Änderungen am Blueprint vor und veröffentlichen automatisch eine neue Version im Bereich. Die neue Version kann dann als Katalogversion hinzugefügt werden.

1. Navigieren Sie zu dem `react-app-blueprint` Projekt, das in erstellt wurde [Tutorial: Eine React-Anwendung erstellen und aktualisieren](#).
2. Öffnen Sie die Entwicklungsumgebung, die in erstellt wurde [Tutorial: Eine React-Anwendung erstellen und aktualisieren](#).
  - a. Wählen Sie im Navigationsbereich Code und dann Dev Environments aus.
  - b. Suchen Sie in der Tabelle nach der Entwicklungsumgebung und wählen Sie dann Öffnen in AWS Cloud9 (im Browser) aus.
3. Als der Blueprint-Release-Workflow ausgeführt wurde, wurde die Blueprint-Version durch Aktualisierung der Datei inkrementiert. `package.json` Übernehmen Sie diese Änderung, indem Sie den folgenden Befehl im Terminal ausführen: AWS Cloud9

```
git pull
```

4. Navigieren Sie zu dem `static-assets` Ordner, indem Sie den folgenden Befehl ausführen:

```
cd /projects/react-app-blueprint/static-assets
```

- Erstellen Sie eine `hello-world.txt` Datei im `static-assets` Ordner, indem Sie den folgenden Befehl ausführen:

```
touch hello-world.txt
```

AWS Cloud9 ist auf einer Linux-basierten Plattform aufgebaut. Wenn Sie ein Windows-Betriebssystem verwenden, können Sie stattdessen den folgenden Befehl verwenden:

```
echo > hello-world.txt
```

- Doppelklicken Sie in der linken Navigationsleiste auf die `hello-world.txt` Datei, um sie im Editor zu öffnen, und fügen Sie den folgenden Inhalt hinzu:

```
Hello, world!
```

Speichern Sie die Datei.

- Stellen Sie sicher, dass Sie über die neuesten Änderungen verfügen, und übertragen Sie die Änderungen anschließend in das CodeCatalyst Quell-Repository des Blueprints, indem Sie die folgenden Befehle ausführen:

```
git pull
```

```
git add .
```

```
git commit -m "prettier setup"
```

```
git push
```

Durch das Übertragen der Änderungen wurde der Release-Workflow gestartet, der die neue Version des Blueprints automatisch im Space veröffentlicht.

## Schritt 6: Aktualisieren Sie die veröffentlichte Katalogversion des Blueprints auf die neue Version

Nachdem ein Blueprint zum Erstellen eines neuen Projekts verwendet oder auf bestehende Projekte angewendet wurde, können Sie den Blueprint weiterhin als Blueprint-Autor aktualisieren. In diesem Schritt nehmen Sie Änderungen am Blueprint vor und ändern die Katalogversion des Blueprints.

1. Navigieren Sie in der CodeCatalyst Konsole zurück zum Bereich.
2. Wählen Sie Einstellungen und dann Blueprints aus.
3. Wählen Sie react-app-blueprint und wählen Sie dann Katalogversion verwalten aus.
4. Wählen Sie die neue Version und dann Speichern aus.

## Schritt 7: Aktualisieren Sie das Projekt mit der neuen Blueprint-Version

Eine neue Version ist jetzt im Blueprint-Katalog des Spaces verfügbar. Als Blueprint-Benutzer können Sie die Version für das in erstellte Projekt aktualisieren. [Schritt 4: Projekt mit Blueprint erstellen](#) Dadurch wird sichergestellt, dass Sie über die neuesten Änderungen und Korrekturen verfügen, die zur Einhaltung der bewährten Verfahren erforderlich sind.

1. Navigieren Sie in der CodeCatalyst Konsole zu dem react-app-project Projekt, das in erstellt wurde [Schritt 4: Projekt mit Blueprint erstellen](#).
2. Wählen Sie im Navigationsbereich die Option Blueprints aus.
3. Wählen Sie im Infofeld Blueprint aktualisieren aus.
4. Im Bereich Codeänderungen auf der rechten Seite können Sie die Änderungen `hello-world.txt` und `package.json` Aktualisierungen sehen.
5. Wählen Sie Update anwenden.

Wenn Sie Update anwenden auswählen, wird im Projekt eine Pull-Anfrage mit den Änderungen aus der aktualisierten Blueprint-Version erstellt. Um die Aktualisierungen am Projekt vorzunehmen, müssen Sie den Pull Request zusammenführen. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#) und [Eine Pull-Anfrage zusammenführen](#).

1. Suchen Sie in der Blueprints-Tabelle nach dem Blueprint. Wählen Sie in der Spalte Status die Option Ausstehende Pull-Anfrage und dann den Link zum offenen Pull-Request aus.
2. Überprüfe den Pull Request und wähle dann Merge aus.



3. Wählen Sie Fast forward merge, um die Standardwerte beizubehalten, und wählen Sie dann Merge aus.

## Schritt 8: Sehen Sie sich die Änderungen im Projekt an

Änderungen am Blueprint sind jetzt danach [Schritt 7: Aktualisieren Sie das Projekt mit der neuen Blueprint-Version](#) in Ihrem Projekt verfügbar. Als Blueprint-Benutzer können Sie die Änderungen im Quell-Repository anzeigen.

1. Wählen Sie im Navigationsbereich Quell-Repositories und dann den Namen des Quell-Repositories aus, das bei der Erstellung des Projekts erstellt wurde.
2. Unter Dateien können Sie die `hello-world.txt` Datei anzeigen, die in [Schritt 5: Blueprint aktualisieren](#) erstellt wurde.
3. Wählen Sie die `hello-world.txt`, um den Inhalt der Datei anzuzeigen.

Das Lebenszyklusmanagement bietet Blueprint-Autoren die Möglichkeit, den Softwareentwicklungszyklus jedes Projekts, das einen bestimmten Blueprint enthält, zentral zu verwalten. Wie in diesem Tutorial gezeigt, können Sie Aktualisierungen an den Blueprint weitergeben, die dann von Projekten übernommen werden können, die den Blueprint verwendet haben, um ein neues Projekt zu erstellen oder ihn auf ein vorhandenes Projekt angewendet haben. Weitere Informationen finden Sie unter [Arbeiten mit Lebenszyklusmanagement als Blueprint-Autor](#).

## Arbeiten mit Lebenszyklusmanagement als Blueprint-Autor

Mit dem Lebenszyklusmanagement können Sie eine große Anzahl von Projekten aus einer einzigen gemeinsamen Quelle bewährter Methoden synchronisieren. Dies skaliert die Verbreitung von Korrekturen und die Wartung einer beliebigen Anzahl von Projekten über ihren gesamten Lebenszyklus der Softwareentwicklung. Das Lebenszyklusmanagement optimiert interne Kampagnen, Sicherheitsfixes, Audits, Laufzeit-Upgrades, Änderungen an bewährten Methoden und anderen Wartungsmethoden, da diese Standards an einem Ort definiert und bei der Veröffentlichung neuer Standards automatisch zentral auf dem neuesten Stand gehalten werden.

Wenn eine neue Version Ihres Blueprints veröffentlicht wird, werden alle Projekte, die diesen Blueprint enthalten, aufgefordert, auf die neueste Version zu aktualisieren. Als Blueprint-Autor können Sie auch die Version eines bestimmten Blueprints sehen, die jedes Projekt zu Compliance-Zwecken enthält. Wenn es Konflikte in einem vorhandenen Quell-Repository gibt, erstellt das Lebenszyklusmanagement Pull-Anforderungen. Für alle anderen Ressourcen, wie z. B.

Entwicklungsumgebung, erstellen alle Lebenszyklusmanagement-Updates streng neue Ressourcen. Benutzer können diese Pull-Anforderungen entweder zusammenführen oder nicht zusammenführen. Wenn die ausstehenden Pull-Anforderungen zusammengeführt werden, wird die Version des Blueprints, einschließlich der Optionen, die in Ihrem Projekt verwendet werden, aktualisiert. Weitere Informationen zum Arbeiten mit dem Lebenszyklusmanagement als Blueprint-Benutzer finden Sie unter [Verwenden des Lebenszyklusmanagements für bestehende Projekte](#) und [Verwenden des Lebenszyklusmanagements für mehrere Blueprints in einem Projekt](#).

## Themen

- [Testen des Lebenszyklusmanagements](#)
- [Arbeiten mit Zusammenführungsstrategien](#)
- [Arbeiten mit Kontextobjekten](#)

## Testen des Lebenszyklusmanagements

Sie können das Lebenszyklusmanagement Ihres Blueprints lokal testen und die Konfliktlösung zusammenführen. Es wird eine Reihe von Paketen unter dem `synth/` Verzeichnis generiert, die die verschiedenen Phasen einer Lebenszyklusaktualisierung darstellen. Um das Lebenszyklusmanagement zu testen, können Sie den folgenden Yarn-Befehl auf Ihrem Blueprint ausführen:`yarn blueprint: resynth` Weitere Informationen zu Resynthetik und Paketen finden Sie unter [Erneute Synchronisierung](#) und [Generieren von Dateien mit Resynthetik](#).

## Arbeiten mit Zusammenführungsstrategien

### Themen

- [Generieren von Dateien mit Resynthetik](#)
- [Verwenden von Zusammenführungsstrategien](#)
- [Angaben von Dateien für Aktualisierungen des Lebenszyklusmanagements](#)
- [Schreiben von Zusammenführungsstrategien](#)

## Generieren von Dateien mit Resynthetik

Erneute Synchronisierung kann den von einem Blueprint erstellten Quellcode mit Quellcode zusammenführen, der zuvor von demselben Blueprint generiert wurde, sodass Änderungen an einem Blueprint an bestehende Projekte weitergegeben werden können. Zusammenführungen werden von der `resynth()` Funktion über Blueprint-Ausgabepakete hinweg ausgeführt. Resynthesis generiert

zunächst drei Pakete, die verschiedene Aspekte des Blueprints und des Projektstatus darstellen. Er kann manuell lokal mit dem `yarn blueprint:resynth` Befehl ausgeführt werden, der die Pakete erstellt, falls sie noch nicht vorhanden sind. Wenn Sie manuell mit den Paketen arbeiten, können Sie das Verhalten der Neusynchronisierung lokal nachahmen und testen. Standardmäßig führen Blueprints nur Resynthesen über die Repositorys unter `aus, src/*` da nur dieser Teil des Pakets in der Regel der Quellkontrolle unterliegt.

- `existing-bundle` – Dieses Paket ist eine Darstellung des vorhandenen Projektstatus. Dies wird künstlich von der -Synthetikberechnung erstellt, um dem Blueprint Kontext darüber zu geben, wofür er in dem Projekt bereitgestellt wird (falls vorhanden). Wenn an dieser Stelle bereits etwas vorhanden ist, wenn die Neusyntax lokal ausgeführt wird, wird es zurückgesetzt und als Mock angesehen. Andernfalls wird es auf den Inhalt des `gesetzancestor-bundle`.
- `ancestor-bundle` – Dies ist das Paket, das die Blueprint-Ausgabe darstellt, wenn sie mit einigen vorherigen Optionen und/oder Version synthetisiert wurde. Wenn dieser Blueprint zum ersten Mal zu einem Projekt hinzugefügt wird, ist der Vorgänger nicht vorhanden, daher ist er auf denselben Inhalt wie der `eingestellteexisting-bundle`. Wenn dieses Paket an dieser Stelle bereits vorhanden ist, wird es lokal als Mock angesehen.
- `proposed-bundle` – Dies ist das Paket, das den Blueprint nachahmt, wenn er mit einigen neuen Optionen und/oder Versionen synthetisiert wurde. Dies ist dasselbe Paket, das von der `synth()` Funktion erstellt wird. Lokal wird dieses Paket immer überschrieben.

Jedes Paket wird während einer Resynthesephase erstellt, auf die über die Blueprint-Klasse unter `zugriffen werden kann this.context.resynthesisPhase.`

- `resolved-bundle` – Dies ist das endgültige Paket, das eine Darstellung dessen darstellt, was für ein CodeCatalyst Projekt gepackt und bereitgestellt wird. Sie können anzeigen, welche Dateien und Unterschiede an die Bereitstellungsmechanismen gesendet werden. Dies ist die Ausgabe der `resynth()` Funktion, die Zusammenführungen zwischen den drei anderen Paketen auflöst.

Die dreifache Zusammenführung wird angewendet, indem der Unterschied zwischen `ancestor-bundle` und `proposed-bundle` angewendet und auf `existing-bundle` angewendet wird, um das `existing-bundle` zu generieren `resolved-bundle`. Alle Zusammenführungsstrategien lösen Dateien in `aufresolved-bundle`. Resyn separiert die Reichweite dieser Pakete mit den Zusammenführungsstrategien des Blueprints während `resynth()` und erzeugt das aufgelöste Paket aus dem Ergebnis.

## Verwenden von Zusammenführungsstrategien

Sie können eine Zusammenführungsstrategie verwenden, die von der Blueprints-Bibliothek angeboten wird. Diese Strategien bieten Möglichkeiten, Dateiausgaben und Konflikte für Dateien zu lösen, die im [Generieren von Dateien mit Resynthetik](#) Abschnitt erwähnt werden.

- `alwaysUpdate` – Eine Strategie, die immer in die vorgeschlagene Datei aufgelöst wird.
- `neverUpdate` – Eine Strategie, die immer in die vorhandene Datei aufgelöst wird.
- `onlyAdd` – Eine Strategie, die in die vorgeschlagene Datei aufgelöst wird, wenn noch keine vorhandene Datei vorhanden ist. Andernfalls wird in die vorhandene Datei aufgelöst.
- `threeWayMerge` – Eine Strategie, die eine dreiseitige Zusammenführung zwischen den vorhandenen, vorgeschlagenen und gemeinsamen Vorgängerdateien durchführt. Die aufgelöste Datei kann Konfliktmarkierungen enthalten, wenn die Dateien nicht sauber zusammengeführt werden können. Der Inhalt der bereitgestellten Dateien muss UTF-8-kodiert sein, damit die Strategie eine aussagekräftige Ausgabe liefert. Die Strategie versucht zu erkennen, ob die Eingabedateien binär sind. Wenn die Strategie einen Zusammenführungskonflikt in einer Binärdatei erkennt, gibt sie immer die vorgeschlagene Datei zurück.
- `preferProposed` – Eine Strategie, die eine dreiseitige Zusammenführung zwischen den vorhandenen, vorgeschlagenen und gemeinsamen Vorgängerdateien durchführt. Diese Strategie löst Konflikte, indem sie die Seite der vorgeschlagenen Datei jedes Konflikts auswählt.
- `preferExisting` – Eine Strategie, die eine dreiseitige Zusammenführung zwischen den vorhandenen, vorgeschlagenen und gemeinsamen Vorgängerdateien durchführt. Diese Strategie löst Konflikte, indem sie die Seite der vorhandenen Datei jedes Konflikts auswählt.

Den Quellcode für die Zusammenführungsstrategien finden Sie im [Open-Source- GitHub Repository](#) .

## Angeben von Dateien für Aktualisierungen des Lebenszyklusmanagements

Während der Neusyntax steuern Blueprints, wie Änderungen in einem vorhandenen Quell-Repository zusammengeführt werden. Möglicherweise möchten Sie jedoch Updates nicht auf jede einzelne Datei in Ihrem Blueprint übertragen. Beispielcode wie CSS-Stylesheets sollen beispielsweise projektspezifisch sein. Die dreiseitige Zusammenführungsstrategie ist die Standardoption, wenn Sie keine andere Strategie angeben. Blueprints können angeben, welche Dateien sie besitzen und welche nicht, indem sie Zusammenführungsstrategien für das Repository-Konstrukt selbst angeben. Blueprints können ihre Zusammenführungsstrategien aktualisieren und die neuesten Strategien können während der Neusynthetik verwendet werden.

```
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/**',
    ],
  },
]);
```

Es können mehrere Zusammenführungsstrategien angegeben werden, und die letzte Strategie hat Vorrang. Nicht erkannte Dateien sind standardmäßig auf eine three-way-merge ähnlich wie Git eingestellt. Es gibt mehrere Zusammenführungsstrategien, die über das MergeStrategiesKonstrukt bereitgestellt werden, aber Sie können Ihre eigenen schreiben. Die bereitgestellten Strategien entsprechen dem [Git-Merge-Strategietreiber](#).

### Schreiben von Zusammenführungsstrategien

Sie können nicht nur eine der bereitgestellten Build-Zusammenführungsstrategien verwenden, sondern auch Ihre eigenen Strategien schreiben. Strategien müssen einer Standardstrategieschnittstelle entsprechen. Sie müssen eine Strategiefunktion schreiben, die Versionen einer Datei aus existing-bundle, proposed-bundle und übernimmt und sie in einer einzigen ancestor-bundle aufgelösten Datei zusammenführt. Beispielsweise:

```
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;

```

Wenn die Dateien nicht vorhanden sind (nicht definiert sind), ist dieser Dateipfad in diesem bestimmten Speicherort-Bundle nicht vorhanden.

Beispiel:

```

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/**',
    ],
  },
],

```

## Arbeiten mit Kontextobjekten

Als Blueprint-Autor können Sie während der Synthetisierung auf Kontext aus dem Blueprint-Projekt zugreifen, um Informationen wie Raum- und Projektnamen oder vorhandene Dateien im Quell-Repository eines Projekts zu erhalten. Sie können auch Details wie die Phase der Neusynthetik abrufen, die der Blueprint generiert. Sie können beispielsweise auf den Kontext zugreifen, um zu erfahren, ob Sie neu synthetisieren, um ein Vorgänger-Bundle oder ein vorgeschlagenes Paket zu generieren. Der vorhandene Codekontext kann dann verwendet werden, um Ihren Code in Ihrem Repository zu transformieren. Sie können beispielsweise Ihre eigene Neusynthetikstrategie

schreiben, um bestimmte Codestandards festzulegen. Die `-Strategie` kann der `-blueprint.ts` Datei für kleine Blueprints hinzugefügt werden, oder Sie können eine separate `-Datei` für `-Strategien` erstellen.

Das folgende Beispiel zeigt, wie Sie Dateien im Kontext eines Projekts finden, einen Workflow Builder festlegen und eine Blueprint-vended Resynmetry-Strategie für eine bestimmte Datei festlegen können:

```
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));
}
}
```

## Entwickeln eines benutzerdefinierten Blueprints

Bevor Sie einen benutzerdefinierten Blueprint veröffentlichen, können Sie ihn so entwickeln, dass er bestimmte Anforderungen erfüllt. Sie können Ihren benutzerdefinierten Blueprint entwickeln und den Blueprint testen, indem Sie bei der Vorschau ein Projekt erstellen. Sie können den

benutzerdefinierten Blueprint so entwickeln, dass er Projektkomponenten wie bestimmten Quellcode, Kontoverbindungen, Workflows, Probleme oder jede andere Komponente enthält, die in erstellt werden kann CodeCatalyst.

### Important

Wenn Sie Blueprint-Pakete aus externen Quellen verwenden möchten, sollten Sie die Risiken berücksichtigen, die mit diesen Paketen verbunden sein können. Sie sind für die benutzerdefinierten Vorlagen verantwortlich, die Sie Ihrem Bereich hinzufügen, und für den Code, den sie generieren.

So entwickeln oder aktualisieren Sie einen benutzerdefinierten Blueprint

1. Setzen Sie Ihre Entwicklungsumgebung fort. Weitere Informationen finden Sie unter [Fortsetzen einer Entwicklungsumgebung](#).

Wenn Sie keine Entwicklungsumgebung haben, müssen Sie zunächst eine erstellen. Weitere Informationen finden Sie unter [Erstellen einer Entwicklungsumgebung](#).

2. Öffnen Sie ein funktionierendes Terminal in Ihrer Entwicklungsumgebung.
3. Wenn Sie sich bei der Erstellung Ihres Blueprints für einen Release-Workflow entschieden haben, wird die neueste Blueprint-Version automatisch veröffentlicht. Rufen Sie die Änderungen ab, um sicherzustellen, dass die `package.json` Datei die erhöhte Version enthält. Verwenden Sie den folgenden Befehl:

```
git pull
```

4. Bearbeiten Sie in der `src/blueprint.ts` Datei die Optionen Ihres benutzerdefinierten Blueprints. Die `Options` Benutzeroberfläche wird vom CodeCatalyst Assistenten dynamisch interpretiert, um eine Auswahlbenutzeroberfläche (UI) zu generieren. Sie können Ihren benutzerdefinierten Blueprint entwickeln, indem Sie Komponenten und unterstützte Tags hinzufügen. Weitere Informationen finden Sie unter [Arbeiten mit einem Assistenten](#), [Umgebungskomponenten](#), [Regionskomponenten](#), [Repository- und Quellcodekomponenten](#), [Workflow-Komponenten](#) und [Komponenten für Entwicklungsumgebungen](#).

Sie können auch das Blueprints SDK und Beispiel-Blueprints anzeigen, um zusätzliche Unterstützung bei der Entwicklung Ihres benutzerdefinierten Blueprints zu erhalten. Weitere Informationen finden Sie im [Open-Source- GitHub Repository](#).



## Arbeiten mit einem Assistenten

Ein Blueprint-Auswahlassistent auf CodeCatalyst wird automatisch von der Options Schnittstelle in der `blueprint.ts` Datei generiert. Der Frontend-Assistent unterstützt Änderungen und Funktionen eines Blueprints mithilfe von [Kommentaren und Tags im JSDOC-Stil](#). Sie können Kommentare und Tags im JSDOC-Stil verwenden, um Aufgaben auszuführen. Sie können beispielsweise den Text auswählen, der über einer Option angezeigt wird, Funktionen wie die Eingabevalidierung aktivieren oder eine Option zusammenklappbar machen. Der Assistent arbeitet, indem er einen abstrakten Syntaxbaum (AST) interpretiert, der von dem - TypeScript Typ aus der -OptionsSchnittstelle generiert wurde. Der Assistent konfiguriert sich automatisch für den so gut wie möglich beschriebenen Typ. Nicht alle Typen werden unterstützt. Andere unterstützte Typen sind die Regionsauswahl und die Umgebungsauswahl.

Im Folgenden finden Sie ein Beispiel für einen Assistenten, der JSDOC-Kommentare und -Tags mit Blueprints verwendetOptions:

```
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[];  
}
```

Der Anzeigename jeder Option der Options Schnittstelle wird camelCase standardmäßig in angezeigt. Klartext im Kommentar zum JSDOC-Stil wird als Text über der Option im Assistenten angezeigt.

## Themen

- [Unterstützte Tags](#)
- [Unterstützte TypeScript Typen](#)
- [Kommunikation mit dem Benutzer während der Synthetisierung](#)

## Unterstützte Tags

Die folgenden JSDOC-Tags werden von einem benutzerdefinierten Blueprint Options im Frontend-Assistenten unterstützt.

### @inlinePolicy ./path/to/policy/file.json

- Erfordert – Option muss vom Typ `seinRole` sein.
- Verwendung – Ermöglicht Ihnen die Kommunikation der Inline-Richtlinien, die eine Rolle benötigt. Es wird erwartet, dass sich der `policy.json` Pfad unter dem Quellcode befindet. Verwenden Sie dieses Tag, wenn Sie eine benutzerdefinierte Richtlinie für eine Rolle benötigen.
- Abhängigkeiten – `blueprint-cli 0.1.12` und höher
- Beispiel – `@inlinePolicy ./deployment-policy.json`

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @inlinePolicy ./path/to/deployment-policy.json
     */
    cdkRole: Role[];
  };
};
```

### @trustPolicy ./path/to/policy/file.json

- Erfordert – Option muss vom Typ `seinRole` sein.
- Verwendung – Ermöglicht Ihnen die Kommunikation der Vertrauensrichtlinien, die eine Rolle benötigt. Es wird erwartet, dass sich der `policy.json` Pfad unter dem Quellcode befindet. Verwenden Sie dieses Tag, wenn Sie eine benutzerdefinierte Richtlinie für eine Rolle benötigen.
- Abhängigkeiten – `blueprint-cli 0.1.12` und höher
- Beispiel – `@trustPolicy ./trust-policy.json`

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @trustPolicy ./path/to/trust-policy.json
     */
    cdkRole: Role[];
  };
};
```

### @validationRegex Regex-Ausdruck

- Erfordert – Option muss eine Zeichenfolge sein.
- Verwendung – Führt eine Eingabevalidierung für die Option mithilfe des angegebenen Regex-Ausdrucks durch und zeigt an @validationMessage.
- Beispiel – @validationRegex /^[a-zA-Z0-9\_]+\$/
- Empfehlung – Verwendung mit @validationMessage. Die Validierungsmeldung ist standardmäßig leer.

### @validationMessage-Zeichenfolge

- Erfordert - @validationRegex oder andere Fehler, um die Nutzung zu überprüfen.
- Verwendung – Zeigt die Validierungsmeldung bei @validation\* Fehlern an.
- Beispiel – @validationMessage Must contain only upper and lowercase letters, numbers, and underscores.
- Empfehlung – Verwendung mit @validationMessage. Die Validierungsmeldung ist standardmäßig leer.

### Zusammengeklappter boolescher Wert (optional)

- Erforderlich – N/A
- Verwendung – Boolescher Wert, mit dem eine Unteroption zusammenklappbar sein kann. Wenn die reduzierte Anmerkung vorhanden ist, ist ihr Standardwert „true“. Wenn Sie den Wert auf setzen, wird ein herunterklappbarer Abschnitt @collapsed false erstellt, der zunächst geöffnet ist.
- Beispiel – @collapsed true

### @displayName-Zeichenfolge

- Erforderlich – N/A
- Verwendung – Anzeigename der Änderungsoption. Erlaubt andere Formate als camelCase für den Anzeigenamen.
- Beispiel – @displayName Blueprint Name

### @displayName-Zeichenfolge

- Erforderlich – N/A
- Verwendung – Anzeigename der Änderungsoption. Erlaubt andere Formate als [camelCase](#) für den Anzeigenamen.
- Beispiel – @displayName Blueprint Name

### @defaultEntropy-Nummer

- Erfordert – Option muss eine Zeichenfolge sein.
- Verwendung – Fügt der Option eine zufällige alphanumerische Zeichenfolge mit einer angegebenen Länge an.
- Beispiel – @defaultEntropy 5

### @placeholder string (optional)

- Erforderlich – N/A
- Verwendung – Ändert den Standardplatzhalter für Textfelder.
- Beispiel – @placeholder type project name here

### @textArea-Nummer (optional)

- Erforderlich – N/A
- Verwendung – Konvertiert Zeichenfolgeneingaben in eine Textbereichskomponente für größere Textabschnitte. Das Hinzufügen einer Zahl definiert die Anzahl der Zeilen. Der Standardwert ist fünf Zeilen.
- Beispiel – @textArea 10

### boolescher Wert bei @hidden (optional)

- Erforderlich – N/A
- Verwendung – Versteckt die Datei vom Benutzer, es sei denn, die Validierungsprüfung schlägt fehl. Der Standardwert ist „true“.
- Beispiel – @hidden

### @Button boolean (optional)

- Erforderlich – N/A
- Verwendung – Die Anmerkung muss sich auf einer booleschen Eigenschaft befinden. Fügt eine Schaltfläche hinzu, die bei Auswahl als wahr synthetisiert wird. Kein Schalter.
- Beispiel – `buttonExample: boolean;`

```
/**
 * @button
 */
buttonExample: boolean;
```

### Boolescher @showName-Wert (optional)

- Erforderlich – N/A
- Verwendung – Kann nur für einen Kontoverbindungstyp verwendet werden. Zeigt die Eingabe des ausgeblendeten Namens an. Standardeinstellung: `default_environment`.
- Beispiel – `@showName true`

```
/**
 * @showName true
 */
accountConnection: AccountConnection<{
    ...
}>;
```

### @showEnvironmentType boolean (optional)

- Erforderlich – N/A

- Verwendung – Kann nur für einen Kontoverbindungstyp verwendet werden. Zeigt das Dropdown-Menü für ausgeblendete Umgebungstypen an. Alle Verbindungen sind standardmäßig auf `eingestelltproduction`. Optionen sind Nicht-Produktion oder Produktion.
- Beispiel – `@showEnvironmentType true`

```
/**
 * @showEnvironmentType true
 */
accountConnection: AccountConnection<{
    ...
}>;
```

### `@forceDefault` boolean (optional)

- Erforderlich – N/A
- Verwendung – Verwendet den vom Blueprint-Autor bereitgestellten Standardwert anstelle des Werts, der zuvor vom Benutzer verwendet wurde.
- Beispiel – `forceDefaultExample: any;`

```
/**
 * @forceDefault
 */
forceDefaultExample: any;
```

### `@erfordert blueprintName`

- Erforderlich – kommentiert die Options Schnittstelle
- Verwendung – Warnt den Benutzer, als Anforderung für den aktuellen Blueprint `blueprintName` auf das Projekt anzuwenden.
- Beispiel – `@requires '@amazon-codecatalyst/blueprints.blueprint-builder'`

```
/*
 * @requires '@amazon-codecatalyst/blueprints.blueprint-builder'
 */
export interface Options extends ParentOptions {
    ...
}
```

## Unterstützte TypeScript Typen

Die folgenden TypeScript Typen werden von einem benutzerdefinierten Blueprint Options im Frontend-Assistenten unterstützt.

### Zahl

- Erfordert – Option muss vom Typ `number` sein.
- Verwendung – Generieren Sie ein Zahleneingabefeld.
- Beispiel – `age: number`

```
{  
  age: number  
  ...  
}
```

### String

- Erfordert – Option muss vom Typ `string` sein.
- Verwendung – Generieren Sie ein Zeichenfolgeneingabefeld.
- Beispiel – `name: string`

```
{  
  age: string  
  ...  
}
```

### Zeichenfolgenliste

- Erfordert – Option muss vom Typ `boolean` sein.
- Verwendung – Generieren Sie ein Kontrollkästchen.
- Beispiel – `isProduction: boolean`

```
{  
  isProduction: boolean  
  ...  
}
```

```
}
```

## Kabel

- Erfordert – Option muss eine Vereinigung von drei oder weniger Zeichenfolgen sein.
- Verwendung – Generieren Sie ein ausgewähltes Optionsfeld.

### Note

Wenn vier oder mehr Elemente vorhanden sind, wird dieser Typ als Dropdown-Menü dargestellt.

- Beispiel – `color: 'red' | 'blue' | 'green'`

```
{  
  color: 'red' | 'blue' | 'green'  
  ...  
}
```

## Dropdown

- Erfordert – Option muss eine Vereinigung von vier oder mehr Zeichenfolgen sein.
- Verwendung – Generieren Sie ein Dropdown-Menü.
- Beispiel – `runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'`

```
{  
  runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'  
  ...  
}
```

## Erweiterbarer Abschnitt

- Erfordert – Option muss ein Objekt sein.
- Verwendung – Generieren Sie einen erweiterbaren Abschnitt. Die Optionen im Objekt werden im erweiterbaren Abschnitt des Assistenten verschachtelt.
- Beispiel –



```
{
  expandableSectionTitle: {
    nestedString: string;
    nestedNumber: number;
  }
}
```

## Tupel

- Erfordert – Option muss vom Typ sein `Tuple`.
- Nutzung – Generieren Sie eine kostenpflichtige Schlüssel-Wert-Eingabe.
- Beispiel – `tuple: Tuple[string, string]>`

```
{
  tuple: Tuple[string, string]>;
  ...
}
```

## Tupelliste

- Erfordert – Option muss ein Array vom Typ sein `Tuple`.
- Verwendung – Generieren Sie eine Tupellisteneingabe.
- Beispiel – `tupleList: Tuple[string, string]>[]`

```
{
  tupleList: Tuple[string, string]>[];
  ...
}
```

## Selector

- Erfordert – Option muss vom Typ sein `Selector`.
- Verwendung – Generieren Sie eine Dropdownliste von Quell-Repositorys oder -Vorlagen, die auf ein Projekt angewendet werden.
- Beispiel – `sourceRepo: Selector<SourceRepository>`

```
{
  sourceRepo: Selector<SourceRepository>;
  sourceRepoOrAdd: Selector<SourceRepository | string>;
  blueprintInstantiation: Selector<BlueprintInstantiation>;
  ...
}
```

## Mehrfachauswahl

- Erfordert – Option muss vom Typ sein `Selector`.
- Verwendung – Generieren Sie eine Multi-Auswahl-Eingabe.
- Beispiel – `multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>`

```
{
  multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>;
  ...
}
```

## Kommunikation mit dem Benutzer während der Synthetisierung

Als Blueprint-Autor können Sie Benutzern nur über Validierungsnachrichten hinaus mitteilen. Beispielsweise kann ein Leerzeichen eine Kombination von Optionen anzeigen, die einen Blueprint erzeugen, der nicht klar ist. Benutzerdefinierte Vorlagen unterstützen die Möglichkeit, Fehlermeldungen durch Aufrufen der `-Synthetik` an Benutzer zurückzugeben. Der Basisentwurf implementiert eine `throwSynthesisError(...)` Funktion, die eine klare Fehlermeldung erwartet. Sie können die Nachricht wie folgt aufrufen:

```
//blueprint.ts
this.throwSynthesisError({
  name: BlueprintSynthesisErrorTypes.BlueprintSynthesisError,
  message: 'hello from the blueprint! This is a custom error communicated to the user.'
})
```

## Umgebungskomponenten

Der benutzerdefinierte Blueprint-Assistent wird dynamisch von der `Options` Schnittstelle generiert, die über den Assistenten bereitgestellt wird. Blueprints unterstützen das Generieren von Benutzeroberflächenkomponenten (UI) aus exponierten Typen.

## So importieren Sie Umgebungskomponenten von Amazon CodeCatalyst Blueprints

Fügen Sie in Ihrer `-blueprint.ts` Datei Folgendes hinzu:

```
import {...} from '@amazon-codecatalyst/codecatalyst-environments'
```

### Themen

- [Erstellen von Entwicklungsumgebungen](#)
- [Beispiele für Mock-Schnittstellen](#)

### Erstellen von Entwicklungsumgebungen

Das folgende Beispiel zeigt, wie Sie Ihre Anwendung in der Cloud bereitstellen:

```
export interface Options extends ParentOptions {  
  ...  
  myNewEnvironment: EnvironmentDefinition{  
    thisIsMyFirstAccountConnection: AccountConnection{  
      thisIsARole: Role['lambda', 's3', 'dynamo'];  
    };  
  };  
}
```

Die Schnittstelle generiert eine UI-Komponente, die nach einer neuen Umgebung (`myNewEnvironment`) mit einer einzigen Kontoverbindung (`fragtthisIsMyFirstAccountConnection`). Eine Rolle für die Kontoverbindung (`thisIsARole`) wird auch mit `['lambda', 's3', 'dynamo']` als die mindestens erforderlichen Rollenfunktionen generiert. Nicht alle Benutzer haben Kontoverbindungen, daher sollten Sie nach dem Fall suchen, dass ein Benutzer kein Konto oder kein Konto mit einer Rolle verbindet. Rollen können auch mit `@inlinePolicies` kommentiert werden. Weitere Informationen finden Sie unter [@inlinePolicy ./path/to/policy/file.json](#).

Die Umgebungskomponente erfordert ein `name` und `environmentType`. Der folgende Code ist die mindestens erforderliche Standardform:

```
{  
  ...  
  "myNewEnvironment": {  
    "name": "myProductionEnvironment",
```

```

    "environmentType": "PRODUCTION"
  },
}

```

Die UI-Komponente fordert Sie dann zur Eingabe verschiedener Felder auf. Wenn Sie die Felder ausfüllen, erhält der Blueprint eine vollständig erweiterte Form. Es kann hilfreich sein, den vollständigen Mock zu Test- und Entwicklungszwecken in die `defaults.json` Datei aufzunehmen.

## Beispiele für Mock-Schnittstellen

### Einfache Mock-Schnittstelle

```

{
  ...
  "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"
        ]
      }
    }
  }
}

```

### Komplexe Mock-Schnittstelle

```

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'];
};
};
}

```

## Mock-Schnittstelle abschließen

```

{
  ...
  "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"
    ]
  }
}
```

```
  },  
}
```

## Secrets-Komponenten

Secrets können in verwendet werden CodeCatalyst , um sensible Daten zu speichern, auf die in Workflows verwiesen werden kann. Sie können Ihrem benutzerdefinierten Blueprint ein Secret hinzufügen und in Ihrem Workflow darauf verweisen. Weitere Informationen finden Sie unter [Arbeiten mit Secrets](#).

So importieren Sie den Amazon CodeCatalyst -Blueprints-Regionstyp

Fügen Sie in Ihrer `-blueprint.ts` Datei Folgendes hinzu:

```
import { Secret, SecretDefinition } from '@amazon-codecatalyst/blueprint-  
component.secrets'
```

### Themen

- [Erstellen eines Secrets](#)
- [Verweisen auf ein Secret in einem Workflow](#)

### Erstellen eines Secrets

Im folgenden Beispiel wird eine UI-Komponente erstellt, die den Benutzer auffordert, einen Secret-Wert und eine optionale Beschreibung einzugeben:

```
export interface Options extends ParentOptions {  
  ...  
  mySecret: SecretDefinition;  
}  
  
export class Blueprint extends ParentBlueprint {  
  constructor(options_: Options) {  
    new Secret(this, options.secret);  
  }  
}
```

Die Secret-Komponente erfordert eine name. Der folgende Code ist die mindestens erforderliche Standardform:

```
{
  ...
  "secret": {
    "name": "secretName"
  },
}
```

## Verweisen auf ein Secret in einem Workflow

Die folgende Beispielvorlage erstellt ein Secret und einen Workflow, der auf den Secret-Wert verweist. Weitere Informationen finden Sie unter [Verweisen auf ein Secret in einem Workflow](#).

```
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(),  
  );  
}
```

Weitere Informationen zur Verwendung von Secrets in CodeCatalyst finden Sie unter [Arbeiten mit Secrets](#).

## Regionskomponenten

Der Regionstyp kann der `Options` Schnittstelle Ihres benutzerdefinierten Blueprints hinzugefügt werden, um eine Komponente im Blueprint-Assistenten zu generieren. Sie können eine oder mehrere AWS-Regionen eingeben. Der Region-Typ kann aus Ihrem Basis-Blueprint in Ihrer `-blueprint.tsDatei` importiert werden. Weitere Informationen finden Sie unter [AWS-Regionen](#).

So importieren Sie den Amazon CodeCatalyst -Blueprints-Regionstyp

Fügen Sie in Ihrer `-blueprint.tsDatei` Folgendes hinzu:

```
import { Region } from '@amazon-codecatalyst/blueprints.blueprint'
```

Der Regionstypparameter ist ein Array von AWS-Regionencodes, aus denen Sie wählen können, oder Sie können verwenden, `*` um alle unterstützten AWS-Regionen einzubeziehen.

### Themen

- [Anmerkungen](#)
- [Beispiele für Regionskomponenten](#)

## Anmerkungen

JSDoc-Tags können jedem Feld in der `Options` Schnittstelle hinzugefügt werden, um anzupassen, wie ein Feld im Assistenten angezeigt wird und sich verhält. Für den Regionstyp werden die folgenden Tags unterstützt:

- Die `@displayName`Anmerkung kann verwendet werden, um die Bezeichnung des Felds im Assistenten zu ändern.

Beispiel: `@displayName AWS Region`

- Die `@placeholder`Anmerkung kann verwendet werden, um den Platzhalter der `Select/Multiselect`-Komponente zu ändern.

Beispiel: `@placeholder Choose AWS Region`

## Beispiele für Regionskomponenten

### Auswählen einer Region aus einer angegebenen Liste

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Region
   */
  region: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>;
}
```

### Auswählen einer oder mehrerer Regionen aus einer angegebenen Liste

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

### Auswählen eines AWS-egions

```
export interface Options extends ParentOptions {
```

```
...
/**
 * @displayName Region
 */
region: Region<['*']>;
}
```

Auswählen einer oder mehrerer Regionen aus einer angegebenen Liste

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

## Repository- und Quellcodekomponenten

Ein Repository wird von Amazon CodeCatalyst zum Speichern von Code verwendet. Das Repository verwendet einen Namen als Eingabe. Die meisten Komponenten werden in einem Repository gespeichert, z. B. Quellcode Dateien, Workflows und andere Komponenten wie verwaltete Entwicklungsumgebungen (MDE). Die Quell-Repository-Komponente exportiert auch Komponenten, die für die Verwaltung von Dateien und statischen Komponenten verwendet werden. Für Repositories gelten Namensbeschränkungen. Weitere Informationen finden Sie unter [Quell-Repositoryen in CodeCatalyst](#).

```
const repository = new SourceRepository(this, {
  title: 'my-new-repository-title',
});
```

So importieren Sie Amazon CodeCatalyst -Blueprints-Repository- und Quellcodekomponenten

Fügen Sie in Ihrer `-blueprint.tsDatei` Folgendes hinzu:

```
import {...} from '@caws-blueprint-component/caws-source-repositories'
```

## Themen

- [Hinzufügen einer Datei](#)
- [Hinzufügen einer generischen Datei](#)

- [Kopieren von Dateien](#)
- [Ausrichtung auf mehrere Dateien](#)
- [Erstellen eines neuen Repositorys und Hinzufügen von Dateien](#)

## Hinzufügen einer Datei

Sie können mit dem `SourceFile` Konstrukt eine Textdatei in ein Repository schreiben. Der Vorgang ist einer der häufigsten Anwendungsfälle und verwendet ein Repository, einen Dateipfad und Textinhalte. Wenn der Dateipfad nicht in einem Repository vorhanden ist, erstellt die Komponente alle erforderlichen Ordner.

```
new SourceFile(repository, `path/to/my/file/in/repo/file.txt`, 'my file contents');
```

### Note

Wenn Sie zwei Dateien an denselben Speicherort innerhalb desselben Repositorys schreiben, überschreibt die letzte Implementierung die vorherige. Sie können das Feature verwenden, um generierten Code zu schichten, und es ist besonders nützlich, um sich über den Code zu erweitern, den die benutzerdefinierten Vorlagen möglicherweise generiert haben.

## Hinzufügen einer generischen Datei

Sie können beliebige Bits in Ihr Repository schreiben. Sie können aus einem Puffer lesen und das `File` Konstrukt verwenden.

```
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());
```

## Kopieren von Dateien

Sie können mit generiertem Code beginnen, indem Sie Startercode kopieren und einfügen und dann mehr Code über diese Basis generieren. Platzieren Sie den Code im `static-assets` Verzeichnis und zielen Sie dann mit dem `StaticAsset` Konstrukt auf diesen Code ab. Der Pfad beginnt in diesem Fall immer im Stammverzeichnis des `static-assets` Verzeichnisses.

```
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'
```

Eine Unterklasse von `StaticAsset` ist `SubstitutionAsset`. Die Unterklasse funktioniert genau gleich, aber stattdessen können Sie eine Mustache-Ersetzung über die Datei ausführen. Es kann nützlich sein, um copy-and-replace Stilgenerierung durchzuführen.

Statische Komponentenersetzung verwendet eine Mustache-Templating-Engine, um die statischen Dateien zu rendern, die in das generierte Quell-Repository eingesät werden. Während des Renderns werden Mustache-Templating-Regeln angewendet, was bedeutet, dass alle Werte standardmäßig HTML-kodiert sind. Um HTML ohne Escape-Zeichen zu rendern, verwenden Sie die Tripel-Syntax `{{{name}}}`. Weitere Informationen finden Sie in den [Vorlagenregeln für Mustaches](#).

#### Note

Das Ausführen eines Ersatzes für Dateien, die nicht textinterpretierbar sind, kann zu Fehlern führen.

```
const starterCodeText = new SubstitutionAsset('path/to/file/file.txt').substitute({
  'my_variable': 'subbed value1',
  'another_variable': 'subbed value2'
})
```

## Ausrichtung auf mehrere Dateien

Statische Komponenten unterstützen das Glob-Targeting über eine statische Funktion auf `StaticAsset` und seine Unterklassen namens `findAll(...)`, die eine Liste statischer Komponenten zurückgibt, die mit ihren Pfaden, Inhalten und mehr vorinstalliert sind. Sie können die Liste mit `File` Konstruktionen verketteten, um Inhalte in das `static-assets` Verzeichnis zu kopieren und einzufügen.

```
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());
```

## Erstellen eines neuen Repositorys und Hinzufügen von Dateien

Sie können eine Repository-Komponente verwenden, um ein neues Repository in einem generierten Projekt zu erstellen. Anschließend können Sie dem erstellten Repository Dateien oder Workflows hinzufügen.

```
import { SourceRepository } from '@amazon-codecatalyst/codecatalyst-source-repositories';  
...  
const repository = new SourceRepository(this, { title: 'myRepo' });
```

Das folgende Beispiel zeigt, wie Sie Dateien und Workflows zu einem vorhandenen Repository hinzufügen:

```
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...**/});
```

Durch die Kombination der beiden Codeteile wird ein einzelnes Repository mit dem Namen myRepo mit einer Quelldatei README.md und einem CodeCatalyst Workflow im Stamm erstellt.

## Workflow-Komponenten

Ein Workflow wird von Amazon- CodeCatalyst Projekten verwendet, um Aktionen basierend auf Auslösern auszuführen. Sie können Workflow-Komponenten verwenden, um Workflow-YAML-Dateien zu erstellen und zusammenzustellen. Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#).

So importieren Sie Workflow-Komponenten von Amazon CodeCatalyst Blueprints

Fügen Sie in Ihrer `-blueprint.ts` Datei Folgendes hinzu:

```
import { WorkflowBuilder, Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
```

## Themen

- [Beispiele für Workflow-Komponenten](#)
- [Herstellen einer Verbindung mit einer Umgebung](#)

## Beispiele für Workflow-Komponenten

### WorkflowBuilder Komponente

Sie können eine Klasse verwenden, um eine Workflow-Definition zu erstellen. Die Definition kann einer Workflow-Komponente zum Rendern in einem Repository zugewiesen werden.

```
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),
});
```

## Workflow-Projen-Komponente

Das folgende Beispiel zeigt, wie eine Projen-Komponente verwendet werden kann, um eine Workflow-YAML in ein Repository zu schreiben:

```
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 ...});
```

## Herstellen einer Verbindung mit einer Umgebung

Viele Workflows müssen in einer AWS-Kontoverbindung ausgeführt werden. Workflows können dies verarbeiten, indem Aktionen eine Verbindung zu Umgebungen mit Konto- und Rollennamensspezifikationen herstellen können.

```
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),
});
```

## Komponenten für Entwicklungsumgebungen

Verwaltete Entwicklungsumgebungen (MDE) werden verwendet, um MDE Workspaces in zu erstellen und bereitzustellen CodeCatalyst. Die Komponente generiert eine `-devfile.yaml` Datei. Weitere Informationen finden Sie unter [Einführung in Devfile](#) und [Eine Repository-Dev-Datei für eine Entwicklungsumgebung verschieben](#).

```
new Workspace(this, repository, SampleWorkspaces.default);
```

So importieren Sie Komponenten von Amazon CodeCatalyst -Blueprints-Workspaces

Fügen Sie in Ihrer `-blueprint.ts` Datei Folgendes hinzu:

```
import {...} from '@amazon-codecatalyst/codecatalyst-workspaces'
```

## Gibt Komponenten aus

In können CodeCatalyst Sie Funktionen, Aufgaben, Fehler und andere Arbeiten überwachen, die an Ihrem Projekt beteiligt sind. Jede Aufgabe wird in einem eindeutigen Datensatz aufbewahrt, der als Problem bezeichnet wird. Jedes Problem kann eine Beschreibung, einen Empfänger, einen Status und andere Eigenschaften haben, nach denen Sie suchen, gruppieren und filtern können. Sie können Ihre Probleme mithilfe der Standardansichten anzeigen oder eigene Ansichten mit benutzerdefinierter Filterung, Sortierung oder Gruppierung erstellen. Weitere Informationen zu Konzepten im Zusammenhang mit Problemen finden Sie unter [Probleme, Konzepte](#) und [Kontingente für Ausgaben in CodeCatalyst](#).

Die Problemkomponente generiert eine JSON-Darstellung eines Problems. Die Komponente nimmt ein ID-Feld und eine Problemdefinition als Eingabe an.

So importieren Sie Amazon CodeCatalyst -Blueprints, die Komponenten ausgeben

Fügen Sie in Ihrer `-blueprint.ts` Datei Folgendes hinzu:

```
import {...} from '@amazon-codecatalyst/blueprint-component.issues'
```

## Themen

- [Beispiele für Problemkomponenten](#)

## Beispiele für Problemkomponenten

### Erstellen eines -Problems

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myFirstIssue', {
  title: 'myFirstIssue',
  content: 'This is an example issue.',
});
```

### Erstellen eines Problems mit hoher Priorität

```
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 ...});
```

### Erstellen eines Problems mit niedriger Priorität mit Labels

```
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'],
```

```
});
```

## Arbeiten mit Blueprint-Tools und CLI

Die [Blueprint-CLI](#) bietet Tools zur Verwaltung und Arbeit mit Ihren benutzerdefinierten Blueprints.

### Themen

- [Arbeiten mit Blueprint-Tools](#)
- [Image-Upload-Tool](#)

### Arbeiten mit Blueprint-Tools

So arbeiten Sie mit den Blueprint-Tools

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Setzen Sie Ihre Entwicklungsumgebung fort. Weitere Informationen finden Sie unter [Fortsetzen einer Entwicklungsumgebung](#).

Wenn Sie keine Entwicklungsumgebung haben, müssen Sie zunächst eine erstellen. Weitere Informationen finden Sie unter [Erstellen einer Entwicklungsumgebung](#).

3. Führen Sie in einem funktionierenden Terminal den folgenden Befehl aus, um die Blueprint-CLI zu installieren:

```
npm install -g @amazon-codecatalyst/blueprint-util.cli
```

4. Importieren Sie in der `blueprint.ts` Datei die Tools, die Sie verwenden möchten, im folgenden Format:

```
import { <tooling-function-name> } from '@amazon-codecatalyst/blueprint-util/cli/lib/<tooling-folder-name>/<tooling-file-name>;
```

#### Tip

Sie können zu [CodeCatalyst blueprints GitHub repository](#), um den Namen der Tools zu finden, die Sie verwenden möchten.

Wenn Sie das Bild-Upload-Tool verwenden möchten, fügen Sie Ihrem Skript Folgendes hinzu:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/  
image-upload-tool/upload-image-to-aws';
```

## Beispiele

- Wenn Sie die Veröffentlichungsfunktion verwenden möchten, fügen Sie Ihrem Skript Folgendes hinzu:

```
import { publish } from '@amazon-codecatalyst/blueprint-util.cli/lib/publish/  
publish';
```

- Wenn Sie das Bild-Upload-Tool verwenden möchten, fügen Sie Ihrem Skript Folgendes hinzu:

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/  
image-upload-tool/upload-image-to-aws';
```

## 5. Rufen Sie die Funktion auf.

### Beispiele:

- Wenn Sie die Veröffentlichungsfunktion verwenden möchten, fügen Sie Ihrem Skript Folgendes hinzu:

```
await publish(logger, config.publishEndpoint, {<your publishing options>});
```

- Wenn Sie das Bild-Upload-Tool verwenden möchten, fügen Sie Ihrem Skript Folgendes hinzu:

```
const {imageUrl, imageName} = await uploadImagePublicly(logger, 'path/to/  
image')););
```

## Image-Upload-Tool

Das Image-Upload-Tool bietet Ihnen die Möglichkeit, Ihr eigenes Image in einen S3-Bucket in Ihrem AWS-Konto hochzuladen und dieses Image dann öffentlich hinter zu verteilen CloudFront. Das Tool verwendet einen Bildpfad im lokalen Speicher (und optional den Bucket-Namen) als Eingabe und gibt die URL an das öffentlich verfügbare Bild zurück. Weitere Informationen finden Sie unter [Was ist Amazon CloudFront?](#) und [Was ist Amazon S3?](#)

## So arbeiten Sie mit dem Image-Upload-Tool

1. Klonen Sie das [Open-Source-Blueprints- GitHub Repository](#), das Zugriff auf das Blueprints SDK und Beispiel-Blueprints bietet. Führen Sie in einem funktionierenden Terminal den folgenden Befehl aus:

```
git clone https://github.com/aws/codecatalyst-blueprints.git
```

2. Führen Sie den folgenden Befehl aus, um zum Blueprints- GitHub Repository zu navigieren:

```
cd codecatalyst-blueprints
```

3. Führen Sie den folgenden Befehl aus, um Abhängigkeiten zu installieren:

```
yarn && yarn build
```

4. Führen Sie den folgenden Befehl aus, um sicherzustellen, dass die neueste Blueprint-CLI-Version installiert ist:

```
yarn upgrade @amazon-codecatalyst/blueprint-util.cli
```

5. Melden Sie sich mit dem S3-Bucket, in den Sie Ihr Image hochladen möchten, beim AWS-Konto an. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI](#) und [Anmelden über die AWS-Befehlszeilenschnittstelle](#).

6. Führen Sie den folgenden Befehl aus dem Stammverzeichnis Ihres CodeCatalyst Repositories aus, um mit der Blueprint-CLI zum Verzeichnis zu navigieren:

```
cd packages/utils/blueprint-cli
```

7. Führen Sie den folgenden Befehl aus, um Ihr Bild in einen S3-Bucket hochzuladen:

```
yarn blueprint upload-image-public <./path/to/your/image>  
      <optional:optional-bucket-name>
```

Es wird eine URL zu Ihrem Image generiert. Die URL ist nicht sofort verfügbar, da die Bereitstellung der CloudFront Verteilung einige Zeit dauert. Überprüfen Sie den Verteilungsstatus, um den neuesten Bereitstellungsstatus zu erhalten. Weitere Informationen finden Sie unter [Arbeiten mit Verteilungen](#).

## Snapshot-Tests

Generierte Snapshot-Tests für mehrere Konfigurationen Ihres Blueprints werden unterstützt.

Blueprints unterstützen [Snapshot-Tests](#) auf Konfigurationen, die Sie als Blueprint-Autor bereitgestellt haben. Die Konfigurationen sind partielle Überschreibungen, die zusätzlich zur Datei `defaults.json` im Stammverzeichnis eines Blueprints zusammengeführt werden. Wenn Snapshot-Tests aktiviert und konfiguriert sind, synthetisiert der Build- und Testprozess die angegebenen Konfigurationen und überprüft, ob sich die synthetisierten Ausgaben nicht gegenüber dem Referenz-Snapshot geändert haben. Informationen zum Anzeigen des Snapshot-Testcodes finden Sie im [CodeCatalyst Blueprints-GitHub Repository](#).

So aktivieren Sie Snapshot-Tests

1. Aktualisieren Sie in der `.projenrc.ts` Datei das Eingabeobjekt auf `ProjenBlueprint` mit den Dateien, die Sie mit Snapshots erstellen möchten. Beispielsweise:

```
{
  ....
  blueprintSnapshotConfiguration: {
    snapshotGlobs: ['**', '!environments/**', '!aws-account-to-environment/**'],
  },
}
```

2. Synthetisieren Sie den Blueprint neu, um TypeScript Dateien in Ihrem Blueprint-Projekt zu erstellen. Bearbeiten Sie die Quelldateien nicht, da sie von Projen verwaltet und neu generiert werden. Verwenden Sie den folgenden Befehl:

```
yarn projen
```

3. Navigieren Sie zum `src/snapshot-configurations` Verzeichnis, um die `default-config.json` Datei mit einem leeren Objekt anzuzeigen. Aktualisieren oder ersetzen Sie die Datei durch eine oder mehrere Ihrer eigenen Testkonfigurationen. Jede Testkonfiguration wird dann mit der `defaults.json` Datei des Projekts zusammengeführt, synthetisiert und beim Testen mit Snapshots verglichen. Verwenden Sie den folgenden Befehl, um zu testen:

```
yarn test
```

Wenn Sie zum ersten Mal einen Testbefehl verwenden, wird die folgende Meldung angezeigt: `Snapshot Summary > NN snapshots written from 1 test suite`. Nachfolgende Testläufe überprüfen, ob sich die synthetisierte Ausgabe nicht gegenüber den Snapshots geändert hat, und zeigen die folgende Meldung an: `Snapshots: NN passed, NN total`.

Wenn Sie Ihren Blueprint absichtlich ändern, um eine andere Ausgabe zu erzeugen, führen Sie den folgenden Befehl aus, um die Referenz-Snapshots zu aktualisieren:

```
yarn test:update
```

Snapshots erwarten, dass synthetisierte Ausgaben zwischen jeder Ausführung konstant sind. Wenn Ihr Blueprint Dateien generiert, die variieren, müssen Sie diese Dateien aus dem Snapshot-Test ausschließen. Aktualisieren Sie das `blueprintSnapshotConfiguration` Objekt Ihres `ProjenBlueprint` Eingabeobjekts, um die `snapshotGlobs` Eigenschaft hinzuzufügen. Die `-snapshotGlobs` Eigenschaft ist ein Array von [Globs](#), das bestimmt, welche Dateien beim Snapshooting ein- oder ausgeschlossen werden.

#### Note

Es gibt eine Standardliste von Globs. Wenn Sie Ihre eigene Liste angeben, müssen Sie die Standardeinträge möglicherweise explizit zurückgeben.

## Veröffentlichen eines benutzerdefinierten Blueprints

Benutzerdefinierte Vorlagen bieten Vorschau Pakete als Ergebnis einer erfolgreichen Synthetisierung. Das Projektpaket stellt den Quellcode, die Konfiguration und die Ressourcen in einem Projekt dar und wird von CodeCatalyst Bereitstellungs-API-Operationen für die Bereitstellung in einem Projekt verwendet. Wenn Sie mit der Entwicklung Ihres benutzerdefinierten Blueprints fortfahren möchten, führen Sie den Blueprint-Synthetisierungsprozess erneut aus. Weitere Informationen finden Sie unter [Konzepte für benutzerdefinierte Vorlagen](#).

#### Important

Wenn Sie Blueprint-Pakete aus externen Quellen verwenden möchten, sollten Sie die Risiken berücksichtigen, die mit diesen Paketen verbunden sein können. Sie sind für die benutzerdefinierten Vorlagen verantwortlich, die Sie Ihrem Bereich hinzufügen, und für den Code, den sie generieren.

## Themen

- [Anzeigen und Veröffentlichen einer Vorschauversion eines benutzerdefinierten Blueprints](#)
- [Anzeigen und Veröffentlichen einer normalen Version eines benutzerdefinierten Blueprints](#)
- [Veröffentlichen und Anwenden eines benutzerdefinierten Blueprints in bestimmten Bereichen und Projekten](#)

## Anzeigen und Veröffentlichen einer Vorschauversion eines benutzerdefinierten Blueprints

Sie können eine Vorschauversion Ihres benutzerdefinierten Blueprints in Ihrem Bereich veröffentlichen, wenn Sie sie dem Blueprints-Katalog Ihres Bereichs hinzufügen möchten. Auf diese Weise können Sie den Blueprint als Benutzer anzeigen, bevor Sie dem Katalog die Nicht-Vorschauversion hinzufügen. Mit der Vorschauversion können Sie veröffentlichen, ohne eine tatsächliche Version zu benötigen. Wenn Sie beispielsweise an einer `0.0.1` Version arbeiten, können Sie eine Vorschauversion veröffentlichen und hinzufügen, sodass neue Updates für eine zweite Version veröffentlicht und als hinzugefügt werden können `0.0.2`.

Nachdem Sie Änderungen vorgenommen haben, erstellen Sie das Paket Ihres benutzerdefinierten Blueprints neu, indem Sie die `-package.json` Datei ausführen, und zeigen Sie eine Vorschau Ihrer Änderungen an.

So zeigen Sie eine Vorschauversion eines benutzerdefinierten Blueprints an und veröffentlichen sie

1. Setzen Sie Ihre Entwicklungsumgebung fort. Weitere Informationen finden Sie unter [Fortsetzen einer Entwicklungsumgebung](#).
2. Öffnen Sie ein funktionierendes Terminal in Ihrer Entwicklungsumgebung.
3. (Optional) Installieren Sie in einem funktionierenden Terminal die erforderlichen Abhängigkeiten für Ihr Projekt, wenn Sie sie noch nicht installiert haben. Verwenden Sie den folgenden Befehl:

```
yarn
```

4. (Optional) Wenn Sie Änderungen an der `.project.ts` Datei vorgenommen haben, generieren Sie die Konfiguration Ihres Projekts neu, bevor Sie Ihren Blueprint erstellen und in der Vorschau anzeigen. Verwenden Sie den folgenden Befehl:

```
yarn projen
```

5. Erstellen Sie Ihren benutzerdefinierten Blueprint mit dem folgenden Befehl neu und zeigen Sie eine Vorschau an. Verwenden Sie den folgenden Befehl:



```
yarn blueprint:preview
```

Navigieren Sie zu dem bereitgestellten `See this blueprint at:` Link, um eine Vorschau Ihrer benutzerdefinierten Vorlage anzuzeigen. Überprüfen Sie, ob die Benutzeroberfläche, einschließlich Text, je nach Konfiguration wie erwartet angezeigt wird. Wenn Sie Ihren benutzerdefinierten Blueprint ändern möchten, können Sie die `blueprint.ts` Datei bearbeiten, den Blueprint neu synchronisieren und dann erneut eine Vorschauversion veröffentlichen. Weitere Informationen finden Sie unter [Erneute Synchronisierung](#).

6. (Optional) Sie können eine Vorschauversion Ihres benutzerdefinierten Blueprints veröffentlichen, die dann dem Blueprint-Katalog Ihres Raums hinzugefügt werden kann. Navigieren Sie zum `Enable version [preview version number] at:` Link, um eine Vorschauversion in Ihrem Bereich zu veröffentlichen.

Sie können die Projekterstellung emulieren, ohne ein Projekt in erstellen zu müssen CodeCatalyst. Verwenden Sie den folgenden Befehl, um Ihr Projekt zu synthetisieren:

```
yarn blueprint:synth
```

Im `synth/synth.[options-name]/proposed-bundle/` Ordner wird ein Blueprint generiert. Weitere Informationen finden Sie unter [Synthesis](#).

Wenn Sie Ihren benutzerdefinierten Blueprint aktualisieren, verwenden Sie stattdessen den folgenden Befehl, um Ihr Projekt neu zu synthetisieren:

```
yarn blueprint:resynth
```

Im `synth/synth.[options-name]/proposed-bundle/` Ordner wird ein Blueprint generiert. Weitere Informationen finden Sie unter [Erneute Synchronisierung](#).

Nachdem Sie Ihre Vorschauversion veröffentlicht haben, können Sie den Blueprint hinzufügen, sodass die Raummitglieder ihn verwenden können, um neue Projekte zu erstellen oder in vorhandenen Projekten anzuwenden. Weitere Informationen finden Sie unter [Einen benutzerdefinierten Blueprint zu einem Space-Katalog hinzufügen](#).

## Anzeigen und Veröffentlichen einer normalen Version eines benutzerdefinierten Blueprints

Nachdem Sie mit der Entwicklung und Vorschau Ihres benutzerdefinierten Blueprints fertig sind, können Sie die neue Version anzeigen und veröffentlichen, die Sie dem Blueprints-Katalog Ihres Raums hinzufügen möchten. Der beim Erstellen eines Projekts generierte Release-Workflow veröffentlicht automatisch Änderungen, die per Push übertragen werden. Wenn Sie die Workflow-Generierung beim Erstellen des Blueprints deaktiviert haben, wird der Blueprint nicht automatisch zur Verfügung gestellt, um dem Blueprint-Katalog Ihres Raums hinzugefügt zu werden. Sie können Ihren benutzerdefinierten Blueprint auch nach der Ausführung eines `yarn` Befehls in Ihrem Bereich veröffentlichen.

So zeigen Sie einen benutzerdefinierten Blueprint an und veröffentlichen ihn

1. Setzen Sie Ihre Entwicklungsumgebung fort. Weitere Informationen finden Sie unter [Fortsetzen einer Entwicklungsumgebung](#).
2. Öffnen Sie ein funktionierendes Terminal in Ihrer Entwicklungsumgebung.
3. • Wenn Sie sich beim Erstellen Ihres Blueprints von der Generierung des Release-Workflows abgemeldet haben, verwenden Sie den folgenden Befehl:

```
yarn blueprint:release
```

Sie können immer noch zu dem `See this blueprint at:` Link navigieren, der bereitgestellt wird, um Ihren benutzerdefinierten Blueprint anzuzeigen.

Veröffentlichen Sie die aktualisierte Version Ihres benutzerdefinierten Blueprints, die dann dem Blueprint-Katalog Ihres Raums hinzugefügt werden kann. Navigieren Sie zum `Enable version [release version number] at:` Link, um die neueste Version in Ihrem Bereich zu veröffentlichen.

- Wenn Sie sich bei der Erstellung Ihres Blueprints für einen Release-Workflow entschieden haben, wird die neueste Blueprint-Version automatisch veröffentlicht, wenn Änderungen übertragen werden. Verwenden Sie die folgenden Befehle:

```
git add .
```

```
git commit -m "commit message"
```

```
git push
```

Nachdem Sie Ihre normale Version veröffentlicht haben, können Sie den Blueprint hinzufügen, sodass die Raummitglieder ihn verwenden können, um neue Projekte zu erstellen oder in vorhandenen Projekten anzuwenden. Weitere Informationen finden Sie unter [Einen benutzerdefinierten Blueprint zu einem Space-Katalog hinzufügen](#).

## Veröffentlichen und Anwenden eines benutzerdefinierten Blueprints in bestimmten Bereichen und Projekten

Standardmäßig werden die `blueprint:release` Befehle `blueprint:preview` und in dem CodeCatalyst Bereich veröffentlicht, in dem Sie den Blueprint erstellt haben. Wenn Sie mehrere Enterprise-Räume haben, können Sie auch in diesen Bereichen dieselbe Vorlage anzeigen und veröffentlichen. Sie können einen Blueprint auch auf ein vorhandenes Projekt eines anderen Bereichs anwenden.

So veröffentlichen oder wenden Sie einen benutzerdefinierten Blueprint in einem bestimmten Bereich an

1. Setzen Sie Ihre Entwicklungsumgebung fort. Weitere Informationen finden Sie unter [Fortsetzen einer Entwicklungsumgebung](#).
2. Öffnen Sie ein funktionierendes Terminal in Ihrer Entwicklungsumgebung.
3. (Optional) Installieren Sie die erforderlichen Abhängigkeiten für Ihr Projekt, wenn Sie sie noch nicht installiert haben. Verwenden Sie den folgenden Befehl:

```
yarn
```

4. Verwenden Sie das `--spaceTag`, um eine Vorschau oder normale Version in einem bestimmten Bereich zu veröffentlichen. Beispielsweise:

- ```
yarn blueprint:preview --space my-awesome-space # publishes under a "preview" version tag to 'my-awesome-space'
```

Beispielausgabe:

```
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'
```

### Beispielausgabe:

```
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
```

Verwenden Sie die `--project`, um eine Vorschauversion eines benutzerdefinierten Blueprints auf ein vorhandenes Projekt in einem bestimmten Bereich anzuwenden. Beispielsweise:

```
yarn blueprint:preview --space my-awesome-space --project my-project # previews blueprint application to an existing project
```

### Beispielausgabe:

```
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
```

## Details, Versionen und Projekte eines benutzerdefinierten Blueprints anzeigen

Du kannst die von deinem Space veröffentlichten benutzerdefinierten Blueprints einsehen, einschließlich der Details, Versionen und der Projekte, die mit dem Blueprint erstellt wurden oder den Blueprint anwenden.

### Themen

- [Die benutzerdefinierten Blueprints eines Spaces anzeigen](#)
- [Projekte anzeigen, die mit einem benutzerdefinierten Blueprint erstellt wurden, oder Projekte anwenden](#)

## Die benutzerdefinierten Blueprints eines Spaces anzeigen

Um die benutzerdefinierten Blueprints eines Spaces anzusehen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie einen benutzerdefinierten Blueprint anzeigen möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus, um die Space-Blueprints anzuzeigen. Die folgenden Details werden in der Tabelle angezeigt:
  - Name — Name des benutzerdefinierten Blueprints.
  - Katalogstatus — Gibt an, ob der benutzerdefinierte Blueprint im Blueprint-Katalog des Spaces veröffentlicht wurde.
  - Letzte Version — Die neueste Version des benutzerdefinierten Blueprints.
  - Letzte Änderung — Das Datum, an dem der Raum-Blueprint zuletzt aktualisiert wurde.

## Projekte anzeigen, die mit einem benutzerdefinierten Blueprint erstellt wurden, oder Projekte anwenden

Um Projekte anzuzeigen, die mit einem benutzerdefinierten Blueprint erstellt wurden oder diesen angewendet haben

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie einen benutzerdefinierten Blueprint anzeigen möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie in der Tabelle Space-Blueprints den Namen eines benutzerdefinierten Blueprints aus, um die Projekte anzuzeigen, die Blueprint verwenden, und Projekte, die keine Blueprint-Tabellen verwenden.

## Hinzufügen und Entfernen eines benutzerdefinierten Blueprints in einem Bereich

Nachdem Sie einen benutzerdefinierten Blueprint in Ihrem Space veröffentlicht haben, kann er dem Blueprint-Katalog Ihres Bereichs hinzugefügt werden. Ein benutzerdefinierter Blueprint kann auch aus dem Blueprint-Katalog Ihres Spaces entfernt werden, wenn Sie nicht mehr möchten, dass er zum Erstellen neuer Projekte verwendet oder auf bestehende Projekte angewendet wird.

### Themen

- [Einen benutzerdefinierten Blueprint zu einem Space-Katalog hinzufügen](#)
- [Einen benutzerdefinierten Blueprint aus einem Space-Katalog entfernen](#)

### Einen benutzerdefinierten Blueprint zu einem Space-Katalog hinzufügen

Wenn Sie dem Blueprint-Katalog Ihres CodeCatalyst Bereichs einen benutzerdefinierten Blueprint hinzufügen, steht der Blueprint allen Space-Mitgliedern zur Verfügung, um ihn bei der Erstellung eines Projekts oder bei der Anwendung auf ein vorhandenes Projekt zu verwenden. Bevor Sie einen benutzerdefinierten Blueprint zum Blueprint-Katalog des Spaces hinzufügen, muss die Veröffentlichungsberechtigung des Blueprints aktiviert sein. Wenn Sie sich für die Generierung von Workflow-Versionen entschieden haben, sind die Veröffentlichungsberechtigungen standardmäßig aktiviert. Weitere Informationen erhalten Sie unter [Veröffentlichungsberechtigungen für einen benutzerdefinierten Blueprint verwalten](#) und [Veröffentlichen eines benutzerdefinierten Blueprints](#).

Um einen Blueprint zum Blueprint-Katalog des Bereichs hinzuzufügen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Der Blueprint kann nur aus dem Standardzweig des Quell-Repositorys hinzugefügt werden. Wenn Sie den Blueprint für einen Feature-Branch entwickelt haben, führen Sie Ihren Feature-Branch mit den Änderungen am Standard-Branch zusammen. Erstelle eine Pull-Anfrage, um alle Änderungen am Standard-Branch zusammenzuführen. Weitere Informationen finden Sie unter [Mit Pull-Requests in Amazon arbeiten CodeCatalyst](#).
3. Navigieren Sie in der CodeCatalyst Konsole zum Space-Dashboard mit Ihrem benutzerdefinierten Blueprint.
4. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.

5. Wählen Sie den Blueprint-Namen aus, den Sie hinzufügen möchten, und wählen Sie dann Zum Katalog hinzufügen aus. Wenn Sie mehr als eine Version haben, wählen Sie eine Version aus dem Dropdownmenü Katalogversion aus
6. Wählen Sie Speichern.

## Einen benutzerdefinierten Blueprint aus einem Space-Katalog entfernen

### Note

Wenn Sie einen benutzerdefinierten Blueprint aus einem Raumkatalog entfernen, hat dies keine Auswirkungen auf ein Projekt, das anhand des Blueprints erstellt wurde, oder auf ein Projekt, auf das der Blueprint angewendet wurde. Ressourcen des Blueprints werden nicht aus dem Projekt entfernt.

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zum Space-Dashboard mit Ihrem benutzerdefinierten Blueprint
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie den Blueprint-Namen aus, den Sie entfernen möchten, und wählen Sie dann Blueprint aus Katalog entfernen aus.

## Veröffentlichungsberechtigungen für einen benutzerdefinierten Blueprint verwalten

Standardmäßig ist die Genehmigung eines benutzerdefinierten Blueprints aktiviert, wenn während der Projekterstellung eine Workflow-Version generiert wurde. Wenn Veröffentlichungsberechtigungen aktiviert sind, kann der Blueprint im Space veröffentlicht werden. Sie können die Berechtigung deaktivieren, sodass der Blueprint nicht veröffentlicht werden kann. Wenn die Berechtigung deaktiviert ist, wird der Release-Workflow, der bei der Blueprint-Erstellung generiert wird, nicht ausgeführt. Neue Änderungen an einem Blueprint können nur veröffentlicht werden, wenn die Berechtigungen des Blueprints aktiviert sind.

**⚠ Important**

Um die Veröffentlichungsberechtigungen eines Blueprint-Projekts zu aktivieren oder zu deaktivieren, müssen Sie über die Rolle eines Space-Administrators verfügen.

Um die Veröffentlichungsberechtigungen eines Blueprint-Projekts zu verwalten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie die Veröffentlichungsberechtigungen eines benutzerdefinierten Blueprints verwalten möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie den Tab „Berechtigungen für die Veröffentlichung von Projekten“, um die Veröffentlichungsberechtigungen für alle Blueprints Ihres Bereichs einzusehen.
5. Wählen Sie den Blueprint aus, den Sie verwalten möchten, und wählen Sie dann Aktivieren oder Deaktivieren, um die Veröffentlichungsberechtigungen zu ändern. Wenn Sie die Berechtigungen aktivieren, überprüfen Sie die Details zur Änderung der Berechtigungen und wählen Sie dann Blueprint-Veröffentlichung aktivieren aus, um die Änderung zu bestätigen.

## Versionen für einen benutzerdefinierten Blueprint verwalten

Als Blueprint-Autor können Sie die Version verwalten, die Sie im Blueprint-Katalog des Bereichs veröffentlichen möchten. Das Ändern der Katalogversion eines Blueprints hat keine Auswirkungen auf Projekte, die eine andere Blueprint-Version verwenden.

Um eine benutzerdefinierte Blueprint-Version zu verwalten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie die Version eines benutzerdefinierten Blueprints ändern möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie in der Tabelle Space-Blueprints das Optionsfeld für den benutzerdefinierten Blueprint aus, den Sie verwalten möchten.
5. Wählen Sie Katalogversion erstellen und wählen Sie dann eine Version aus dem Dropdownmenü Katalogversion aus.
6. Wählen Sie Speichern.



## Löschen eines veröffentlichten benutzerdefinierten Blueprints oder einer veröffentlichten Version

Wenn Sie die Version eines benutzerdefinierten Blueprints oder den Blueprint selbst aus Ihrem CodeCatalyst Amazon-Bereich löschen, wird Ihr gesamter Zugriff auf die Ressourcen des Blueprint-Projekts oder der Blueprint-Version entfernt. Wenn Sie eine Blueprint-Version oder den Blueprint gelöscht haben, können Projektmitglieder nicht mehr auf Projektressourcen zugreifen, und alle Workflows, die von Quell-Repositorys von Drittanbietern angefordert werden, werden gestoppt.

### Note

Wenn Sie einen Blueprint löschen, hat dies keine Auswirkungen auf ein Projekt, auf das der Blueprint angewendet wurde. Ressourcen des Blueprints werden nicht aus dem Projekt entfernt.

Wenn eine Blueprint-Version im Blueprint-Katalog des Spaces veröffentlicht wird, wählen Sie eine neue Version für den Katalog aus, bevor Sie die veröffentlichte Version löschen.

Um die Katalogversion eines benutzerdefinierten Blueprints zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie die Katalogversion eines benutzerdefinierten Blueprints löschen möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie den Namen des Blueprints mit der Katalogversion, die Sie löschen möchten.
5. Wählen Sie das Optionsfeld für die Katalogversion, die Sie löschen möchten, und wählen Sie dann Version löschen aus.
6. Überprüfen Sie die Details und wählen Sie dann im Dropdownmenü Neue Blueprint-Katalogversion auswählen eine andere Blueprint-Version aus.
7. Geben Sie `eindelete`, um die Auswahl der Blueprint-Katalogversion zu bestätigen.
8. Wählen Sie Löschen.

Wenn eine Blueprint-Version nicht im Blueprint-Katalog des Spaces enthalten ist, können Sie die Version löschen, ohne eine neue Version auszuwählen.

## Um eine benutzerdefinierte Blueprint-Version zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie eine benutzerdefinierte Blueprint-Version löschen möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie den Namen des Blueprints mit der Version, die Sie löschen möchten.
5. Wählen Sie das Optionsfeld für die Version, die Sie löschen möchten, und wählen Sie dann Version löschen aus.
6. Geben Sie `delete` die Eingabetaste ein, um das Löschen der Blueprint-Version zu bestätigen.
7. Wählen Sie Löschen.

Wenn Sie einen Blueprint aus dem Blueprint-Katalog des Spaces löschen, werden alle Versionen des Blueprints gelöscht. Die Projekte des Spaces, die den Blueprint verwenden, sind vom Löschen nicht betroffen.

## Um eine benutzerdefinierte Blueprint-Version zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie einen benutzerdefinierten Blueprint löschen möchten.
3. Wählen Sie im Space-Dashboard die Registerkarte Einstellungen und dann Blueprints aus.
4. Wählen Sie in der Tabelle Space-Blueprints das Optionsfeld für den benutzerdefinierten Blueprint aus, den Sie löschen möchten, und wählen Sie dann Blueprint löschen aus.
5. Geben Sie die Eingabetaste `delete`, um das Löschen des benutzerdefinierten Blueprints zu bestätigen.
6. Wählen Sie Löschen.

## Arbeiten mit Abhängigkeiten und Tools

### Themen

- [Hinzufügen von Abhängigkeiten](#)
- [Umgang mit Nichtübereinstimmungen des Abhängigkeitstyps](#)
- [Verwenden von Yarn und npm](#)

- [Aktualisieren von Tools und Komponenten](#)

## Hinzufügen von Abhängigkeiten

Als Blueprint-Autor müssen Sie Ihrem Blueprint möglicherweise Pakete hinzufügen, z. B. `@amazon-codecatalyst/blueprint-component.environments`. Sie müssen die `projen.ts` Datei mit diesem Paket aktualisieren und dann die Konfiguration Ihres Projekts mit [Projen](#) neu generieren. Projen fungiert als Projektmodell für jede Blueprint-Codebasis, die die Möglichkeit bietet, abwärtskompatible Tooling-Updates zu übertragen, indem die Art und Weise geändert wird, wie das Modell Konfigurationsdateien rendert. Die `package.json` Datei ist eine Datei, die teilweise dem Projen-Modell gehört. Projen bestätigt Abhängigkeitsversionen, die in der `package.json`-Datei enthalten sind, aber andere Optionen müssen aus dem Modell stammen.

So fügen Sie eine Abhängigkeit hinzu und aktualisieren eine **`projenrc.ts`** Datei

1. Navigieren Sie in der `projen.ts` Datei zum Abschnitt `deps`.
2. Fügen Sie die Abhängigkeit hinzu, die Sie in Ihrem Blueprint verwenden möchten.
3. Verwenden Sie den folgenden Befehl, um die Konfiguration Ihres Projekts neu zu generieren:

```
yarn projen && yarn
```

## Umgang mit Nichtübereinstimmungen des Abhängigkeitstyps

Nach einer [Yarn](#)-Aktualisierung wird möglicherweise die folgende Fehlermeldung in Bezug auf einen Repository-Parameter angezeigt:

```
Type 'SourceRepository' is missing the following properties from type  
'SourceRepository': synthesisSteps, addSynthesisStep
```

Der Fehler ist auf eine Nichtübereinstimmung der Abhängigkeit zurückzuführen, die auftritt, wenn eine Komponente auf eine neuere Version einer anderen Komponente angewiesen ist, die vertrauende Komponente jedoch auf eine ältere Version fixiert ist. Der Fehler kann behoben werden, indem alle Ihre Komponenten auf derselben Version basieren, sodass die Version zwischen ihnen synchronisiert wird. Es empfiehlt sich, alle von der Vorlage verkauften Pakete unter derselben neuesten Version (`0.0.x`) zu halten, es sei denn, Sie sind sich sicher, wie Sie die Versionen behandeln. Das folgende Beispiel zeigt, wie die `package.json` Datei so konfiguriert werden kann, dass alle Abhängigkeiten von derselben Version abhängen:

```
...
"@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": "*",
```

Nachdem Sie die Versionen für alle Abhängigkeiten konfiguriert haben, verwenden Sie den folgenden Befehl:

```
yarn install
```

## Verwenden von Yarn und npm

Blueprints verwenden [Yarn](#) für Tools. Die Verwendung von [npm](#) und Yarn führt zu Tooling-Problemen, da die Art und Weise, wie Abhängigkeitsbäume von jedem Baum aufgelöst werden, unterschiedlich ist. Um solche Probleme zu vermeiden, empfiehlt es sich, nur Yarn zu verwenden.

Wenn Sie versehentlich Abhängigkeiten mit npm installiert haben, können Sie die generierte `package-lock.json` Datei entfernen und sicherstellen, dass Ihre `.projenrc.ts` Datei mit den benötigten Abhängigkeiten aktualisiert wird. Sie generieren die Konfiguration Ihres Projekts mit `Projen neu`.

Verwenden Sie Folgendes, um aus dem Modell neu zu generieren:

```
yarn projen
```

Nachdem Sie sichergestellt haben, dass Ihre `.projenrc.ts`-Datei mit den erforderlichen Abhängigkeiten aktualisiert wurde, verwenden Sie den folgenden Befehl:

```
yarn
```

## Aktualisieren von Tools und Komponenten

Gelegentlich sollten Sie Ihre Tools und Komponenten aktualisieren, um neue Funktionen verfügbar zu machen. Es wird empfohlen, alle Komponenten auf derselben Version zu halten, es sei denn, Sie sind sich sicher, wie Sie die Versionen behandeln. Versionen werden zwischen Komponenten

synchronisiert, sodass dieselben Versionen für alle Komponenten eine ordnungsgemäße Abhängigkeit zwischen ihnen sicherstellen.

### Verwenden von Yarn Workspace Monorepo

Verwenden Sie den folgenden Befehl, um Nutzungen und Komponenten aus dem Stammverzeichnis des Repositorys eines benutzerdefinierten Blueprints zu aktualisieren:

```
yarn upgrade @amazon-codecatalyst/*
```

Verwenden Sie den folgenden Befehl, wenn Sie kein Monorepo verwenden:

```
yarn upgrade --pattern @amazon-codecatalyst/*
```

Weitere Optionen, die Sie zum Upgraden von Tools und Komponenten verwenden können:

- Verwenden Sie die `npm-Ansicht@caws-blueprint-component/<some-component>`, um die neueste Version zu erhalten.
- Erhöhen Sie manuell auf die neueste Version, indem Sie die Version in Ihrer `package.json`-Datei festlegen und den folgenden Befehl verwenden:`yarn` Alle Komponenten und Anwendungen sollten dieselbe Version haben.

## Beitragend

Das Blueprints Software Development Kit (SDK) ist eine Open-Source-Bibliothek, zu der Sie beitragen können. Berücksichtigen Sie als Beitragender die Beitragsrichtlinien, das Feedback und die Fehler. Weitere Informationen finden Sie im [Blueprints- GitHub Repository](#).

## Kontingente für Blueprints in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Limits für Blueprints in Amazon CodeCatalyst beschrieben. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

Maximale Anzahl von Blueprints, die pro CodeCatalyst Projekt angewendet werden	100
--	-----

# Quell-Repositoryn in CodeCatalyst

CodeCatalyst Quell-Repositorys sind Git-Repositorys, die in Amazon gehostet werden. CodeCatalyst Sie können Quell-Repositorys verwenden, CodeCatalyst um Ressourcen für ein Projekt sicher zu speichern, zu versionieren und zu verwalten.

Zu den Ressourcen in einem CodeCatalyst Repository können gehören:

- -Documents
- Quellcode
- binäre Dateien

CodeCatalyst verwendet auch das Quell-Repository für ein Projekt, um Konfigurationsinformationen für Ihr Projekt zu speichern, z. B. Workflow-Konfigurationsdateien.

Sie können mehr als ein Quell-Repository in einem CodeCatalyst Projekt haben. Möglicherweise möchten Sie beispielsweise separate Quell-Repositorys für Front-End-Quellcode, Back-End-Quellcode, Dienstprogramme und Dokumentation haben.

Hier ist ein möglicher Workflow für die Arbeit mit Code in Quell-Repositorys, Pull-Requests und Entwicklungsumgebungen in: CodeCatalyst

Mary Major erstellt ein Webanwendungsprojekt CodeCatalyst mithilfe eines Blueprints, der ein Quell-Repository mit Beispielcode erstellt. Sie lädt ihre Freunde Li Juan, Saanvi Sarkar und Jorge Souza ein, mit ihr an dem Projekt zu arbeiten. Li Juan schaut sich den Beispielcode im Quell-Repository an und beschließt, einige schnelle Änderungen vorzunehmen, um dem Code einen Test hinzuzufügen. Li erstellt eine Entwicklungsumgebung, wählt AWS Cloud9 sie als IDE aus und spezifiziert einen neuen Zweig, *Testcode*. Die Entwicklungsumgebung wird geöffnet. Li fügt schnell den Code hinzu, überträgt dann den Branch mit den Änderungen und überträgt ihn in das Quell-Repository. CodeCatalyst Li erstellt dann eine Pull-Anfrage. Im Rahmen der Erstellung dieser Pull-Anfrage fügt Li Jorge Souza und Saanvi Sarkar als Reviewer hinzu, um sicherzustellen, dass der Code überprüft wird.

Bei der Überprüfung des Codes erinnert sich Jorge Souza daran, dass er sein eigenes Projekt-Repository hat, das einen Prototyp der App enthält GitHub, an der sie gerade arbeiten. Er bittet Mary Major, die Erweiterung zu installieren und zu konfigurieren, die es ihm ermöglicht, das GitHub Repository als zusätzliches Quell-Repository mit dem Projekt zu verknüpfen. Mary überprüft

das Repository auf GitHub und arbeitet mit Jorge zusammen, um die GitHub Erweiterung so zu konfigurieren, dass er das GitHub Repository als zusätzliches Quell-Repository für das Projekt verknüpfen kann.

CodeCatalyst Quell-Repositorys unterstützen die Standardfunktionalität von Git und funktionieren mit Ihren vorhandenen Git-basierten Tools. Sie können Personal Access Tokens (PATs) als anwendungsspezifisches Passwort erstellen und verwenden, wenn Sie Quell-Repositorys von einem Git-Client oder integrierten Entwicklungsumgebungen (IDEs) klonen und damit arbeiten. Diese PATs sind mit Ihrer Benutzeridentität verknüpft. CodeCatalyst Weitere Informationen finden Sie unter [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#).

CodeCatalyst Quell-Repositorys unterstützen Pull-Requests. Dies ist eine einfache Möglichkeit für dich und andere Projektmitglieder, Codeänderungen zu überprüfen und zu kommentieren, bevor du sie von einem Branch zum anderen zusammenführst. Sie können die Änderungen in der CodeCatalyst Konsole anzeigen und Codezeilen kommentieren.

Durch Pushes an Branches in einem CodeCatalyst Quell-Repository kann automatisch eine Ausführung in einem Workflow gestartet werden, in dem Änderungen erstellt, getestet und bereitgestellt werden können. Wenn Ihr Quell-Repository als Teil eines Projekts mithilfe einer Projektvorlage erstellt wurde, werden ein oder mehrere Workflows als Teil des Projekts für Sie konfiguriert. Sie können jederzeit zusätzliche Workflows für Repositorys hinzufügen. Die YAML-Konfigurationsdateien für Workflows in einem Projekt werden in den Quell-Repositorys gespeichert, die in der Quellaktion für diese Workflows konfiguriert wurden. Weitere Informationen finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#).

## Themen

- [Konzepte für Quell-Repositoryn](#)
- [Einrichtung für die Arbeit mit Quell-Repositorys](#)
- [Erste Schritte mit CodeCatalyst Quell-Repositorys und dem Single-Page-Anwendungs-Blueprint](#)
- [Arbeiten mit Quell-Repositorys in CodeCatalyst](#)
- [Zusammenarbeit mit Filialen bei Amazon CodeCatalyst](#)
- [Arbeiten mit Dateien in Amazon CodeCatalyst](#)
- [Mit Pull-Requests in Amazon arbeiten CodeCatalyst](#)
- [Mit Commits in Amazon arbeiten CodeCatalyst](#)
- [Kontingente für Quell-Repositorys in CodeCatalyst](#)

# Konzepte für Quell-Repositoryn

Im Folgenden finden Sie einige Konzepte, die Sie bei der Arbeit mit CodeCatalyst Quell-Repositorys kennen sollten.

Themen

- [Projekte](#)
- [Quell-Repositorys](#)
- [Entwicklungsumgebungen](#)
- [Persönliche Zugriffstoken \(PATs\)](#)
- [Branches](#)
- [Standardzweige](#)
- [Commits](#)
- [Pull-Anforderungen](#)
- [Überarbeitungen](#)
- [Workflows](#)

## Projekte

Ein Projekt ist ein Gemeinschaftsprojekt CodeCatalyst, das Entwicklungsteams und Aufgaben unterstützt. Sobald Sie ein Projekt erstellt haben, können Sie Benutzer und Ressourcen hinzufügen, aktualisieren oder entfernen, Ihr Projekt-Dashboard anpassen und den Fortschritt der Arbeit Ihres Teams überwachen. Sie können mehrere Projekte in einem Bereich haben.

Quell-Repositorys sind spezifisch für die Projekte, in denen Sie sie in einem Space erstellen oder verknüpfen. Sie können ein Repository nicht von mehreren Projekten gemeinsam nutzen und Sie können ein Repository nicht mit mehr als einem Projekt in einem Space verknüpfen. Benutzer mit der Rolle Mitwirkender oder Projektadministrator in einem Projekt können mit den Quell-Repositorys interagieren, die diesem Projekt zugeordnet sind, je nach den Berechtigungen, die diesen Rollen gewährt wurden. Weitere Informationen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

## Quell-Repositorys

In einem Quell-Repository speichern Sie Code und Dateien für Ihr Projekt sicher. Es speichert auch den Versionsverlauf Ihrer Dateien. Standardmäßig wird ein Quell-Repository mit den anderen



Benutzern in Ihrem CodeCatalyst Projekt geteilt. Sie können mehr als ein Quell-Repository für ein Projekt haben. Sie können Quell-Repositories für Projekte in erstellen oder ein vorhandenes Quell-Repository verknüpfen CodeCatalyst, das von einem anderen Dienst gehostet wird, sofern dieser Dienst von einer installierten Erweiterung unterstützt wird. Sie können beispielsweise ein GitHub Repository mit einem Projekt verknüpfen, nachdem Sie die GitHub Repository-Erweiterung installiert haben. Weitere Informationen finden Sie unter [Arbeiten mit Quell-Repositories in CodeCatalyst](#) und [Schnellstart: GitHub Repositories verwenden in CodeCatalyst](#).

## Entwicklungsumgebungen

Eine Entwicklungsumgebung ist eine cloudbasierte Entwicklungsumgebung, mit der Sie schnell CodeCatalyst an dem Code arbeiten können, der in den Quell-Repositories Ihres Projekts gespeichert ist. Die in Ihrer Entwicklungsumgebung enthaltenen Projekttools und Anwendungsbibliotheken werden durch eine Dev-Datei im Quell-Repository Ihres Projekts definiert. Wenn Sie keine Dev-Datei in Ihrem Quell-Repository haben, wird automatisch eine Standard-Dev-Datei angewendet. Die Standard-Devdatei enthält Tools für die am häufigsten verwendeten Programmiersprachen und Frameworks. Standardmäßig ist eine Entwicklungsumgebung so konfiguriert, dass sie über einen 2-Core-Prozessor, 4 GB RAM und 16 GiB persistenten Speicher verfügt.

Sie können wählen, ob Sie einen vorhandenen Zweig Ihres Quell-Repositories in Ihre Entwicklungsumgebung klonen möchten, oder Sie können wählen, ob Sie im Rahmen der Erstellung der Entwicklungsumgebung einen neuen Zweig erstellen möchten.

## Persönliche Zugriffstoken (PATs)

Ein Personal Access Token (PAT) ähnelt einem Passwort. Es ist mit Ihrer Benutzeridentität verknüpft und kann in allen Bereichen und Projekten in verwendet CodeCatalyst werden. Sie verwenden PATs, um auf CodeCatalyst Ressourcen zuzugreifen, die integrierte Entwicklungsumgebungen (IDEs) und Git-basierte Quell-Repositories umfassen. PATs repräsentieren Sie in CodeCatalyst und Sie können sie in Ihren Benutzereinstellungen verwalten. Ein Benutzer kann mehr als eine PAT haben. Persönliche Zugriffstoken werden nur einmal angezeigt. Es hat sich bewährt, sie sicher auf Ihrem lokalen Computer zu speichern. Standardmäßig laufen PATs nach einem Jahr ab.

Bei der Arbeit mit integrierten Entwicklungsumgebungen (IDEs) entsprechen PATs einem Git-Passwort. Geben Sie die PAT an, wenn Sie bei der Einrichtung Ihrer IDE für die Arbeit mit einem Git-Repository nach einem Passwort gefragt werden. Weitere Informationen darüber, wie Sie Ihre IDE mit einem Git-basierten Repository verbinden, finden Sie in der Dokumentation zu Ihrer IDE.

## Branches

Ein Branch ist ein Zeiger oder eine Referenz auf einen Commit in Git und in CodeCatalyst. Du kannst Branches verwenden, um deine Arbeit zu organisieren. Sie können beispielsweise Zweige verwenden, um an einer neuen oder anderen Version von Dateien zu arbeiten, ohne dass sich dies auf Dateien in anderen Verzweigungen auswirkt. Sie können Branches verwenden, um neue Funktionen zu entwickeln, eine bestimmte Version Ihres Projekts zu speichern und vieles mehr. Ein Quell-Repository kann einen Zweig oder mehrere Zweige haben. Wenn Sie ein Projekt mithilfe einer Vorlage erstellen, enthält das für das Projekt erstellte Quell-Repository Beispieldateien in einem Zweig namens main. Der Hauptzweig ist der Standardzweig für das Repository.

## Standardzweige

Quell-Repositorys CodeCatalyst haben einen Standard-Branch, unabhängig davon, wie Sie sie erstellen. Wenn Sie ein Projekt mithilfe einer Vorlage erstellen möchten, enthält das für dieses Projekt erstellte Quell-Repository neben Beispielcode, Workflow-Definitionen und anderen Ressourcen auch eine README.md-Datei. Wenn Sie ein Quell-Repository erstellen, ohne eine Vorlage zu verwenden, wird als erster Commit eine README.md-Datei für Sie hinzugefügt, und im Rahmen der Erstellung des Repositorys wird ein Standardbranch für Sie erstellt. Dieser Standardzweig heißt main. Dieser Standardzweig wird als Basis- oder Standardzweig in lokalen Repositorys (Repos) verwendet, wenn Benutzer das Repository klonen. Sie können ändern, welcher Branch als Standard-Branch verwendet wird. Weitere Informationen finden Sie unter [Den Standardzweig für ein Repository anzeigen und ändern](#).

Sie können den Standardzweig für ein Quell-Repository nicht löschen. Die Suchergebnisse enthalten nur Ergebnisse aus dem Standardzweig.

## Commits

Ein Commit ist eine Änderung an einer Datei oder einer Gruppe von Dateien. In der CodeCatalyst Amazon-Konsole speichert ein Commit Ihre Änderungen und überträgt sie in ein Quell-Repository. Der Commit enthält Informationen über die Änderung, einschließlich der Identität des Benutzers, der die Änderung vorgenommen hat, Uhrzeit und Datum der Änderung, den Commit-Titel und alle Nachrichten zu der Änderung. Weitere Informationen finden Sie unter [Mit Commits in Amazon arbeiten CodeCatalyst](#).

Im Kontext eines Quell-Repositorys in CodeCatalyst sind Commits Schnappschüsse der Inhalte und Änderungen an den Inhalten Ihres Repositorys. Du kannst Commits auch Git-Tags hinzufügen, um bestimmte Commits zu identifizieren.

## Pull-Anforderungen

Eine Pull-Anfrage ist die wichtigste Methode, mit der du und andere Benutzer Codeänderungen von einem Branch zum anderen in einem Quell-Repository überprüfen, kommentieren und zusammenführen. Sie können Pull Requests verwenden, um Codeänderungen gemeinsam auf kleinere Änderungen oder Korrekturen, wichtige Funktionserweiterungen oder neue Versionen Ihrer veröffentlichten Software zu überprüfen. In einem Pull Request kannst du die Änderungen zwischen den Quell- und Ziel-Branched oder die Unterschiede zwischen den Versionen dieser Branches überprüfen. Du kannst Kommentare zu einzelnen Codezeilen sowie Kommentare zum gesamten Pull-Request hinzufügen.

### Tip

Während du einen Pull Request erstellst, ist der angezeigte Unterschied der Unterschied zwischen der Spitze des Quell-Branched und der Spitze des Ziel-Branched. Sobald der Pull Request erstellt wurde, besteht der angezeigte Unterschied zwischen der Revision des Pull-Requests, den du ausgewählt hast, und dem Commit, der bei der Erstellung des Pull-Requests der Tip des Ziel-Branched war. Weitere Informationen zu Unterschieden und Merge-Bases in Git finden Sie [git-merge-base](#) in der Git-Dokumentation.

## Überarbeitungen

Eine Revision ist eine aktualisierte Version einer Pull-Anfrage. Jeder Push an den Quell-Branch einer Pull-Anfrage erstellt eine Revision, die die Änderungen enthält, die an den in diesem Push enthaltenen Commits vorgenommen wurden. Du kannst dir die Unterschiede zwischen den Versionen eines Pull-Requests sowie die Unterschiede zwischen den Quell- und Ziel-Branched ansehen. Weitere Informationen finden Sie unter [Mit Pull-Requests in Amazon arbeiten CodeCatalyst](#).

## Workflows

Ein Workflow ist ein automatisiertes Verfahren, das beschreibt, wie Sie Ihren Code als Teil eines CI/CD-Systems (Continuous Integration and Continuous Delivery) erstellen, testen und bereitstellen. Ein Workflow definiert eine Reihe von Schritten oder Aktionen, die während einer Workflow-Ausführung ausgeführt werden sollen. Ein Workflow definiert auch die Ereignisse oder Auslöser, die den Start des Workflows auslösen. Um einen Workflow einzurichten, erstellen Sie mit dem [visuellen Editor oder dem YAML-Editor](#) der CodeCatalyst Konsole eine Workflow-Definitionsdatei.

**i** Tip

Um einen schnellen Überblick darüber zu erhalten, wie Sie Workflows in einem Projekt verwenden könnten, [erstellen Sie ein Projekt mit einem Blueprint](#). Jeder Blueprint stellt einen funktionierenden Workflow bereit, den Sie überprüfen, ausführen und mit dem Sie experimentieren können.

In einem Quell-Repository können auch die Konfigurationsdateien und andere Informationen für Workflows, Benachrichtigungen, Probleme und andere Konfigurationsinformationen für das Projekt gespeichert werden. Die Konfigurationsdateien werden erstellt und im Quell-Repository gespeichert, wenn Sie Ressourcen erstellen, für die Konfigurationsdateien erforderlich sind, oder wenn Sie das Repository als Quellaktion für einen Workflow angeben. Wenn Sie ein Projekt anhand eines Blueprints erstellen, sind die Konfigurationsdateien bereits im Quell-Repository gespeichert, das für Sie als Teil des Projekts erstellt wurde. Diese Konfigurationsinformationen werden in einem Ordner gespeichert, der `.codecatalyst` im Standardzweig Ihres Repositories benannt ist. Immer wenn Sie einen Zweig des Standardzweigs erstellen, erstellen Sie zusätzlich zu allen anderen Dateien und Ordnern in diesem Zweig eine Kopie dieses Ordners und seiner Konfiguration.

## Einrichtung für die Arbeit mit Quell-Repositorys

Wenn Sie mit Quell-Repositorys in Amazon CodeCatalyst auf Ihrem lokalen Computer arbeiten, können Sie Git entweder eigenständig oder in einer unterstützten integrierten Entwicklungsumgebung (IDE) verwenden, um Codeänderungen vorzunehmen und Ihren Code zu pushen und abzurufen. Als bewährte Methode empfehlen wir, die neuesten Versionen von Git und anderer Software zu verwenden.

**i** Note

Wenn Sie Dev Environments verwenden, müssen Sie Git nicht installieren. Eine aktuelle Version von Git ist in Ihrer Entwicklungsumgebung enthalten.

Informationen zur Versionskompatibilität für CodeCatalyst

Komponente	Version
Git	brandneue

## Installieren Sie Git

Um mit Dateien, Commits, Branches und anderen Informationen in Quell-Repositorys von einem Git-Client ohne IDE aus zu arbeiten, installieren Sie Git auf Ihrem lokalen Computer.

Um Git zu installieren, empfehlen wir Websites wie [Git Downloads](#).

## Erstellen Sie ein persönliches Zugriffstoken

Um Quell-Repositorys auf Ihrem lokalen Computer oder in Ihrer bevorzugten IDE zu klonen, müssen Sie ein Personal Access Token (PAT) erstellen.

Um ein Personal Access Token (PAT) zu erstellen

1. Wählen Sie in der oberen Menüleiste Ihr Profil-Badge und dann Meine Einstellungen aus.

### Tip

Du kannst dein Benutzerprofil auch finden, indem du die Mitgliederseite für ein Projekt oder einen Bereich aufrufst und deinen Namen aus der Mitgliederliste auswählst.

2. Geben Sie im Feld PAT-Name einen aussagekräftigen Namen für Ihr PAT ein.
3. Behalten Sie im Feld Ablaufdatum das Standarddatum bei oder wählen Sie das Kalendersymbol, um ein benutzerdefiniertes Datum auszuwählen. Das Ablaufdatum liegt standardmäßig auf einem Jahr ab dem aktuellen Datum.
4. Wählen Sie Erstellen.

Sie können dieses Token auch erstellen, wenn Sie Repository klonen für ein Quell-Repository wählen.

5. Speichern Sie das PAT-Geheimnis an einem sicheren Ort.

### Important

Das PAT-Geheimnis wird nur einmal angezeigt. Sie können es nicht abrufen, nachdem Sie das Fenster geschlossen haben.

# Erste Schritte mit CodeCatalyst Quell-Repositorys und dem Single-Page-Anwendungs-Blueprint

Folgen Sie den Schritten in diesem Tutorial, um zu erfahren, wie Sie mit Quell-Repositorys in Amazon CodeCatalyst arbeiten.

Der schnellste Weg, um mit der Arbeit mit Quell-Repositorys in Amazon zu beginnen, CodeCatalyst besteht darin, ein Projekt mithilfe einer Vorlage zu erstellen. Wenn Sie ein Projekt mithilfe einer Vorlage erstellen, werden Ressourcen für Sie erstellt, einschließlich eines Quell-Repositorys, das Beispielcode enthält. Sie können dieses Repository und dieses Codebeispiel verwenden, um zu lernen, wie Sie:

- Die Quell-Repositorys eines Projekts anzeigen und deren Inhalt durchsuchen
- Erstellen Sie eine Entwicklungsumgebung mit einem neuen Zweig, in dem Sie an Code arbeiten können
- Ändern Sie eine Datei und übertragen Sie Ihre Änderungen und übertragen Sie sie
- Erstelle einen Pull-Request und überprüfe deine Codeänderungen mit anderen Projektmitgliedern
- Sieh dir an, wie der Workflow für dein Projekt die Änderungen automatisch im Quellzweig des Pull-Requests erstellt und testet
- Führe deine Änderungen aus deinem Quell-Branch mit dem Ziel-Branch zusammen und schließe den Pull Request
- Sieh dir an, wie die zusammengeführten Änderungen automatisch erstellt und bereitgestellt werden

Um das Beste aus diesem Tutorial herauszuholen, laden Sie andere zu Ihrem Projekt ein, damit Sie gemeinsam an einem Pull-Request arbeiten können. Sie können sich auch mit zusätzlichen Funktionen vertraut machen CodeCatalyst, z. B. Probleme erstellen und sie einer Pull-Anfrage zuordnen oder Benachrichtigungen konfigurieren und Benachrichtigungen erhalten, wenn der zugehörige Workflow ausgeführt wird. Eine vollständige Erläuterung von finden Sie CodeCatalyst unter [Tutorials für die ersten Schritte](#).

## Ein Projekt mithilfe eines Blueprints erstellen

Das Erstellen eines Projekts ist der erste Schritt, um zusammenarbeiten zu können. Sie können einen Blueprint verwenden, um Ihr Projekt zu erstellen. Dabei wird auch ein Quell-Repository mit Beispielcode und einem Workflow erstellt, der Ihren Code automatisch erstellt und bereitstellt, wenn Sie ihn ändern. In diesem Tutorial führen wir Sie durch ein Projekt, das mit dem einseitigen

Anwendungs-Blueprint erstellt wurde. Sie können jedoch die Verfahren für jedes Projekt mit einem Quell-Repository befolgen. Stellen Sie sicher, dass Sie eine IAM-Rolle auswählen oder eine IAM-Rolle hinzufügen, falls Sie bei der Erstellung des Projekts noch keine haben. Wir empfehlen, dass Sie die CodeCatalystWorkflowDevelopmentRole-*spaceName*ServiceRolle für dieses Projekt verwenden.

Wenn Sie bereits ein Projekt haben, können Sie weitermachen mit [Die Repositories für ein Projekt anzeigen](#).

#### Note

Nur Benutzer mit der Rolle Space-Administrator oder Poweruser können Projekte in erstellen CodeCatalyst. Wenn Sie diese Rolle nicht haben und für dieses Tutorial ein Projekt benötigen, an dem Sie arbeiten können, bitten Sie jemanden mit einer dieser Rollen, ein Projekt für Sie zu erstellen und Sie dem erstellten Projekt hinzuzufügen. Weitere Informationen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

Um ein Projekt mit einem Blueprint zu erstellen

1. Navigieren Sie in der CodeCatalyst Konsole zu dem Bereich, in dem Sie ein Projekt erstellen möchten.
2. Wählen Sie im Space-Dashboard die Option Projekt erstellen aus.
3. Wählen Sie Mit einem Blueprint beginnen aus.
4. Wählen Sie einen Blueprint und dann Weiter aus.
5. Geben Sie unter Geben Sie Ihrem Projekt einen Namen den Namen ein, den Sie Ihrem Projekt zuweisen möchten, sowie die zugehörigen Ressourcennamen. Der Name muss in Ihrem Bereich eindeutig sein.
6. Konfigurieren Sie unter Projektressourcen allgemeine Projektparameter.
7. (Optional) Um Definitionsdateien mit Aktualisierungen auf der Grundlage der von Ihnen ausgewählten Projektparameter anzuzeigen, wählen Sie unter Projektvorschau generieren die Option Code anzeigen oder Workflow anzeigen aus.
8. (Optional) Wählen Sie auf der Karte des Blueprints die Option Details anzeigen aus, um spezifische Details zum Blueprint anzuzeigen, z. B. einen Überblick über die Architektur des Blueprints, die erforderlichen Verbindungen und Berechtigungen sowie die Art der Ressourcen, die der Blueprint erstellt.
9. Wählen Sie Create project (Projekt erstellen) aus.

Weitere Informationen zu Projekt-Blueprints finden Sie unter [Referenz zum Projekt-Blueprint](#)

Die Seite mit der Projektübersicht wird geöffnet, sobald Sie ein Projekt erstellen oder eine Einladung zu einem Projekt annehmen und den Anmeldevorgang abgeschlossen haben. Die Projektübersichtsseite für ein neues Projekt enthält keine offenen Probleme oder Pull-Requests. Du kannst dich optional dafür entscheiden, ein Problem zu erstellen und es dir selbst zuzuweisen. Sie können sich auch dafür entscheiden, eine andere Person zu Ihrem Projekt einzuladen. Weitere Informationen finden Sie unter [Ein Problem erstellen in CodeCatalyst](#) und [Einen Benutzer zu Ihrem Projekt einladen](#).

## Die Repositorys für ein Projekt anzeigen

Als Mitglied eines Projekts können Sie die Quell-Repositorys für das Projekt einsehen. Sie können sich auch dafür entscheiden, zusätzliche Repositorys zu erstellen. Wenn jemand mit der Space-Administratorrolle die GitHub Repository-Erweiterung installiert und konfiguriert hat, können Sie auch Links zu GitHub Repositorys in den für die Erweiterung konfigurierten GitHub Konten hinzufügen. Weitere Informationen finden Sie unter [Erstellen eines Quell-Repositorys](#) und [Schnellstart: GitHub Repositorys verwenden in CodeCatalyst](#).

### Note

**Bei Projekten, die mit dem Blueprint für einseitige Anwendungen erstellt wurden, lautet der Standardname für das Quell-Repository, das den Beispielcode enthält, spa-app.**

Um zu den Quell-Repositorys für ein Projekt zu navigieren

1. Navigieren Sie zu Ihrem Projekt und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie auf der Übersichtsseite für Ihr Projekt das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen.
  - Wählen Sie im Navigationsbereich Code und dann Quell-Repositorys aus. Wählen Sie unter Quell-Repositorys den Namen des Repositorys aus der Liste aus. Sie können die Liste der Repositorys filtern, indem Sie einen Teil des Repository-Namens in die Filterleiste eingeben.
2. Sehen Sie sich auf der Startseite des Repositorys den Inhalt des Repositorys und Informationen zu den zugehörigen Ressourcen an, z. B. die Anzahl der Pull-Requests und Workflows.



Standardmäßig werden die Inhalte für den Standard-Branch angezeigt. Sie können die Ansicht ändern, indem Sie einen anderen Zweig aus der Dropdownliste auswählen.

Die Übersichtsseite für das Repository enthält Informationen zu den Workflows und Pull-Requests, die für die Branches dieses Repositorys und dessen Dateien konfiguriert sind. Wenn Sie das Projekt gerade erst erstellt haben, werden die ersten Workflows zum Erstellen, Testen und Bereitstellen des Codes weiterhin ausgeführt, da sie einige Minuten in Anspruch nehmen. Sie können die zugehörigen Workflows und ihren Status anzeigen, indem Sie die Nummer unter Verwandte Workflows auswählen. Dadurch wird jedoch die Seite Workflows in CI/CD geöffnet. Bleiben Sie für dieses Tutorial auf der Übersichtsseite und erkunden Sie den Code im Repository. Der Inhalt der README .md Datei wird auf dieser Seite unter den Repository-Dateien gerendert. Unter Dateien wird der Inhalt des Standardzweigs angezeigt. Sie können die Dateiansicht so ändern, dass der Inhalt eines anderen Zweigs angezeigt wird, falls Sie einen haben. Der `.codecatalyst` Ordner enthält Code, der für andere Teile des Projekts verwendet wird, z. B. für Workflow-YAML-Dateien.

Um den Inhalt von Ordnern anzuzeigen, klicken Sie auf den Pfeil neben dem Ordernamen, um ihn zu erweitern. Wählen Sie beispielsweise den Pfeil neben, `src` um die Dateien für die einseitige Webanwendung anzuzeigen, die in diesem Ordner enthalten sind. Um den Inhalt einer Datei anzuzeigen, wählen Sie sie in der Liste aus. Dadurch wird Dateien anzeigen geöffnet, in dem Sie den Inhalt mehrerer Dateien durchsuchen können. Sie können auch einzelne Dateien in der Konsole bearbeiten, aber um mehrere Dateien zu bearbeiten, sollten Sie eine Entwicklungsumgebung erstellen.

## Erstellen einer Entwicklungsumgebung

Sie können Dateien in einem Quell-Repository in der CodeCatalyst Amazon-Konsole hinzufügen und ändern. Um effektiv mit mehreren Dateien und Branches arbeiten zu können, empfehlen wir jedoch, eine Entwicklungsumgebung zu verwenden oder das Repository auf Ihren lokalen Computer zu klonen. In diesem Tutorial erstellen wir eine AWS Cloud9 Entwicklungsumgebung mit einem Zweig namens **develop**. Sie können einen anderen Branch-Namen wählen, aber wenn Sie den Branch benennend **develop**, wird automatisch ein Workflow ausgeführt, um Ihren Code zu erstellen und zu testen, wenn Sie später in diesem Tutorial eine Pull-Anfrage erstellen.

### Tip

Wenn Sie sich dafür entscheiden, ein Repository lokal zu klonen, anstatt eine Entwicklungsumgebung zu verwenden oder zusätzlich zu verwenden, stellen Sie sicher,

dass Sie Git auf Ihrem lokalen Computer haben oder dass Ihre IDE Git enthält. Weitere Informationen finden Sie unter [Einrichtung für die Arbeit mit Quell-Repositorys](#).

Um eine Entwicklungsumgebung mit einem neuen Zweig zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung erstellen möchten.
3. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code, dann Quell-Repositorys und dann das Repository auswählen, für das Sie eine Entwicklungsumgebung erstellen möchten.
4. Wählen Sie auf der Repository-Startseite die Option Create Dev Environment aus.
5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
6. Wählen Sie das zu klonende Repository aus, wählen Sie In neuem Zweig arbeiten aus, geben Sie einen Branch-Namen in das Feld Branch-Name ein und wählen Sie im Drop-down-Menü Branch erstellen einen Branch aus, aus dem der neue Branch erstellt werden soll.
7. Fügen Sie optional einen Alias für die Entwicklungsumgebung hinzu.
8. Wählen Sie optional die Schaltfläche „Konfiguration der Entwicklungsumgebung bearbeiten“, um die Rechen-, Speicher- oder Timeout-Konfiguration der Entwicklungsumgebung zu bearbeiten.
9. Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, wird in der Spalte Entwicklungsumgebung der Wert Wird gestartet angezeigt und in der Spalte wird nach Erstellung der Entwicklungsumgebung ausgeführt angezeigt. Es öffnet sich ein neuer Tab mit Ihrer Entwicklungsumgebung in der IDE Ihrer Wahl. Sie können Code bearbeiten und Ihre Änderungen übernehmen und pushen.

Sobald Sie die Entwicklungsumgebung erstellt haben, können Sie Dateien bearbeiten, Ihre Änderungen übernehmen und Ihre Änderungen an den **test** Branch übertragen. Bearbeiten Sie für dieses Tutorial den Inhalt zwischen den `<p>` Tags in der `App.tsx` Datei im `src` Ordner, um den Text zu ändern, der auf der Webseite angezeigt wird. Bestätigen und übertragen Sie Ihre Änderung und kehren Sie dann zur CodeCatalyst Registerkarte zurück.

Um eine Änderung in einer AWS Cloud9 Entwicklungsumgebung vorzunehmen und zu pushen

1. Erweitern Sie in AWS Cloud9 das seitliche Navigationsmenü, um die Dateien zu durchsuchen. Erweitern `src` und öffnen `App.tsx`.

2. Nehmen Sie eine Änderung am Text in den <p> Tags vor.
3. Speichern Sie die Datei und übertragen Sie dann Ihre Änderungen mithilfe des Git-Menüs. Alternativ können Sie Ihre Änderungen im Terminalfenster mit den git push Befehlen git commit und übertragen.

```
git commit -am "Making an example change"  
git push
```

### Tip

Möglicherweise müssen Sie die Verzeichnisse im Terminal in das Git-Repository-Verzeichnis ändern, bevor Sie die Git-Befehle erfolgreich ausführen können.

## Erstellen einer Pull-Anforderung

Sie können Pull Requests verwenden, um Codeänderungen gemeinsam auf kleinere Änderungen oder Korrekturen, wichtige Funktionserweiterungen oder neue Versionen Ihrer veröffentlichten Software zu überprüfen. In diesem Tutorial erstellen Sie eine Pull-Anfrage, um die Änderungen, die Sie am **Test-Branch** vorgenommen haben, im Vergleich zum Main-Branch zu überprüfen. Wenn Sie einen Pull Request in einem Projekt erstellen, das mit einer Vorlage erstellt wurde, wird auch eine Ausführung der zugehörigen Workflows gestartet, falls vorhanden.

Um einen Pull Request zu erstellen

1. Navigiere zu deinem Projekt.
2. Führen Sie eine der folgenden Aktionen aus:
  - Wählen Sie im Navigationsbereich Code, dann Pull Requests und dann Create Pull Request aus.
  - Wählen Sie auf der Repository-Startseite die Option Mehr und dann Pull-Request erstellen aus.
  - Wähle auf der Projektseite die Option Pull-Request erstellen aus.
3. Stellen Sie im Quell-Repository sicher, dass das angegebene Quell-Repository das ist, das den festgeschriebenen Code enthält. Diese Option wird nur angezeigt, wenn Sie den Pull-Request nicht von der Hauptseite des Repositorys aus erstellt haben.


4. Wählen Sie im Ziel-Branch den Branch aus, in den der Code nach der Überprüfung zusammengeführt werden soll.
5. Wählen Sie im Quellzweig den Zweig aus, der den festgeschriebenen Code enthält.
6. Geben Sie im Feld Titel der Pull-Anfrage einen Titel ein, der anderen Benutzern hilft zu verstehen, was überprüft werden muss und warum.
7. (Optional) Geben Sie unter Beschreibung des Pull-Requests Informationen wie einen Link zu Problemen oder eine Beschreibung Ihrer Änderungen ein.

 Tip

Du kannst Beschreibung für mich schreiben wählen, damit CodeCatalyst automatisch eine Beschreibung der in der Pull-Anfrage enthaltenen Änderungen generiert wird. Sie können Änderungen an der automatisch generierten Beschreibung vornehmen, nachdem Sie sie dem Pull Request hinzugefügt haben.

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).

8. (Optional) Wählen Sie unter Probleme die Option Probleme verknüpfen aus und wählen Sie dann entweder ein Problem aus der Liste aus oder geben Sie dessen ID ein. Um die Verknüpfung eines Problems aufzuheben, wählen Sie das Symbol „Verknüpfung aufheben“.
9. (Optional) Wählen Sie unter Erforderliche Prüfer die Option Erforderliche Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Erforderliche Prüfer müssen die Änderungen genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.

 Note

Sie können einen Prüfer nicht gleichzeitig als erforderlichen Prüfer und als optionalen Prüfer hinzufügen. Sie können sich nicht selbst als Prüfer hinzufügen.

10. (Optional) Wählen Sie unter Optionale Prüfer die Option Optionale Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Optionale Prüfer müssen die Änderungen nicht als Anforderung genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.
11. Überprüfe die Unterschiede zwischen den Branches. Der Unterschied, der in einem Pull Request angezeigt wird, besteht in den Änderungen zwischen der Revision im Quell-Branch und der

Merge-Base, bei der es sich um den Head-Commit des Ziel-Branches zum Zeitpunkt der Erstellung des Pull-Requests handelt. Wenn keine Änderungen angezeigt werden, sind die Branches möglicherweise identisch, oder Sie haben denselben Branch sowohl für die Quelle als auch für das Ziel ausgewählt.

12. Wenn Sie überzeugt sind, dass der Pull Request den Code und die Änderungen enthält, die Sie überprüfen möchten, wählen Sie **Create** aus.

#### Note

Nachdem Sie den Pull Request erstellt haben, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @-Zeichen gefolgt vom Namen der Datei verwendest.

Sie können Informationen zu den zugehörigen Workflows anzeigen, die bei der Erstellung dieses Pull-Requests gestartet wurden, indem Sie **Überblick** wählen und dann die Informationen im Bereich **Pull-Request-Details** unter **Workflow-Läufe** überprüfen. Um die Workflow-Ausführung anzuzeigen, wählen Sie die **Ausführung** aus.

#### Tip

Wenn Sie Ihrem Branch einen anderen Namen gegeben haben als **develop**, wird nicht automatisch ein Workflow ausgeführt, um Ihre Änderungen zu erstellen und zu testen. Wenn Sie das konfigurieren möchten, bearbeiten Sie die YAML-Datei für den `onPullRequestBuildAndTestWorkflow`. Weitere Informationen finden Sie unter [Einen Workflow erstellen, bearbeiten und löschen](#).

Sie können diesen Pull-Request kommentieren und andere Projektmitglieder bitten, ihn zu kommentieren. Du kannst dich auch dafür entscheiden, optionale oder erforderliche Prüfer hinzuzufügen oder zu ändern. Du kannst entscheiden, ob du weitere Änderungen am Quell-Branch für das Repository vornehmen möchtest, und dir ansehen, wie diese festgeschriebenen Änderungen zu Revisionen für den Pull Request führen. Weitere Informationen finden Sie unter [Einen Pull-Request überprüfen](#), [Aktualisierung einer Pull-Anfrage](#) [Mit Pull-Requests in Amazon arbeiten CodeCatalyst](#), und [Anzeigen des Status und der Details der Workflow-Ausführung](#).

## Eine Pull-Anfrage zusammenführen

Sobald ein Pull Request geprüft wurde und Genehmigungen von den erforderlichen Prüfern erhalten hat, kannst du seinen Quell-Branch in der Konsole mit dem Ziel-Branch zusammenführen. CodeCatalyst Durch das Zusammenführen eines Pull-Requests wird außerdem eine Ausführung der Änderungen in allen Workflows gestartet, die dem Ziel-Branch zugeordnet sind. In diesem Tutorial führen Sie den Testzweig mit dem Hauptzweig zusammen, wodurch eine Ausführung des `onPushToMainDeployPipelineWorkflows` gestartet wird.

Um eine Pull-Anfrage zusammenzuführen (Konsole)

1. Wählen Sie unter Pull-Anfragen die Pull-Anfrage aus, die Sie im vorherigen Schritt erstellt haben. Wähle in der Pull-Anfrage Merge aus.
2. Wählen Sie aus den verfügbaren Merge-Strategien für den Pull Request aus. Aktiviere oder deaktiviere optional die Option, den Quell-Branch nach dem Zusammenführen des Pull-Requests zu löschen, oder deaktiviere sie und wähle dann Merge. Nach Abschluss der Zusammenführung ändert sich der Status des Pull-Requests in Zusammengeführt und wird nicht mehr in der Standardansicht der Pull Requests angezeigt. In der Standardansicht werden Pull Requests mit dem Status Offen angezeigt. Du kannst einen zusammengeführten Pull Request immer noch ansehen, aber du kannst ihn nicht genehmigen oder seinen Status ändern.

### Note

Wenn die Schaltfläche Zusammenführen nicht aktiv ist oder du die Bezeichnung Nicht zusammenführbar siehst, hat entweder ein erforderlicher Prüfer den Pull Request noch nicht genehmigt, oder der Pull Request kann nicht in der CodeCatalyst Konsole zusammengeführt werden. Ein Prüfer, der eine Pull-Anfrage nicht genehmigt hat, wird in der Übersicht im Bereich Pull-Request-Details durch ein Uhrensymbol gekennzeichnet. Wenn alle erforderlichen Reviewer den Pull Request genehmigt haben, aber die Schaltfläche Zusammenführen immer noch nicht aktiv ist, liegt möglicherweise ein Mergekonflikt vor, oder Sie haben das Speicherkontingent für den Speicherplatz überschritten. Sie können Merge-Konflikte für den Ziel-Branch in einer Entwicklungsumgebung lösen, die Änderungen per Push übertragen und dann den Pull-Request zusammenführen, oder Sie können Konflikte lösen und lokal zusammenführen und dann den Commit, der die Zusammenführung enthält, per Push übertragen. CodeCatalyst Weitere Informationen finden Sie unter [Eine Pull-Anfrage zusammenführen \(Git\)](#) und in Ihrer Git-Dokumentation.

## Den bereitgestellten Code anzeigen

Jetzt ist es an der Zeit, sich den ursprünglich bereitgestellten Code anzusehen, der sich im Standard-Branch befand, sowie Ihre zusammengeführten Änderungen, sobald sie automatisch erstellt, getestet und bereitgestellt wurden. Dazu können Sie zur Übersichtsseite für das Repository zurückkehren und die Zahl neben dem Symbol für zugehörige Workflows auswählen oder im Navigationsbereich CI/CD und dann Workflows auswählen.

Um den bereitgestellten Code anzuzeigen

1. Erweitern Sie unter Workflows in `onPushToMainDeployPipeline` den Bereich Letzte Ausführungen.

### Note

Dies ist der Standardname des Workflows für Projekte, die mit dem Blueprint für einseitige Anwendungen erstellt wurden.

2. Bei der letzten Ausführung handelt es sich um die Ausführung, die mit Ihrem zusammengeführten Pull-Request-Commit an den `main` Branch gestartet wurde. Wahrscheinlich wird der Status In Bearbeitung angezeigt. Wählen Sie einen erfolgreich abgeschlossenen Lauf aus der Liste aus, um die Details dieses Laufs zu öffnen.
3. Wählen Sie Variablen. Kopieren Sie den Wert für `AppUrl`. Dies ist die URL für die bereitgestellte einseitige Webanwendung. Öffnen Sie einen neuen Browser-Tab und fügen Sie den Wert ein, um den erstellten und bereitgestellten Code anzuzeigen. Lassen Sie die Registerkarte geöffnet.
4. Kehren Sie zur Liste der Workflow-Ausführungen zurück und warten Sie, bis die letzte Ausführung abgeschlossen ist. Wenn dies der Fall ist, kehren Sie zu der Registerkarte zurück, die Sie geöffnet haben, um die Webanwendung anzuzeigen, und aktualisieren Sie Ihren Browser. Sie sollten die Änderungen sehen, die Sie in Ihrer zusammengeführten Pull-Anfrage vorgenommen haben.

## Bereinigen von Ressourcen

Nachdem du dich mit der Arbeit mit einem Quell-Repository und einer Pull-Anfrage befasst hast, möchtest du vielleicht alle Ressourcen entfernen, die du nicht benötigst. Sie können Pull-Requests nicht löschen, aber Sie können sie schließen. Du kannst alle Branches löschen, die du erstellt hast.

Wenn Sie das Quell-Repository oder das Projekt nicht mehr benötigen, können Sie diese Ressourcen auch löschen. Weitere Informationen finden Sie unter [Löschen eines Quell-Repositorys](#) und [Löschen eines Projekts in Amazon CodeCatalyst](#).

## Arbeiten mit Quell-Repositorys in CodeCatalyst

In einem Quell-Repository speichern Sie Code und Dateien für Ihr Projekt sicher. Es speichert auch deinen Quellverlauf, vom ersten Commit bis zu den letzten Änderungen. Wenn Sie einen Blueprint wählen, der ein Quell-Repository enthält, enthält dieses Repository auch die Konfigurationsdateien und andere Informationen für Workflows und Benachrichtigungen für das Projekt. Diese Konfigurationsinformationen werden in einem Ordner namens `.codecatalyst` gespeichert.

Sie können ein Quell-Repository erstellen, indem Sie CodeCatalyst entweder ein Projekt mit einem Blueprint erstellen, der im Rahmen der Erstellung eines Projekts ein Quell-Repository erstellt, oder indem Sie ein Quell-Repository in einem vorhandenen Projekt erstellen. Projektbenutzer sehen automatisch die Repositorys, die Sie für ein Projekt erstellen, und können sie verwenden. Sie können sich auch dafür entscheiden, ein auf Ihrem Projekt gehostetes Git-Repository GitHub zu verknüpfen. Wenn Sie dies tun, können Ihre Projektbenutzer das verknüpfte Repository in der Liste der Repositorys für das Projekt anzeigen und darauf zugreifen.

### Note

Bevor Sie das Repository verknüpfen können, müssen Sie die Erweiterung für den Dienst installieren, der es hostet. Sie können ein archiviertes Repository nicht verknüpfen. Sie können zwar ein leeres Repository verknüpfen, aber Sie können es CodeCatalyst erst verwenden, wenn Sie es mit einem ersten Commit initialisiert haben, das einen Standard-Branch erstellt. Weitere Informationen finden Sie unter [Eine Erweiterung installieren](#).

Standardmäßig wird ein Quell-Repository mit anderen Mitgliedern Ihres CodeCatalyst Amazon-Projekts geteilt. Sie können zusätzliche Quell-Repositorys für ein Projekt erstellen oder Repositorys mit dem Projekt verknüpfen. Alle Mitglieder eines Projekts können Dateien und Ordner in den Quell-Repositorys des Projekts anzeigen, hinzufügen, bearbeiten und löschen.

Um schnell an Code in einem Quell-Repository zu arbeiten, können Sie eine Entwicklungsumgebung erstellen, die ein bestimmtes Repository klonet und in dieses verzweigt, wo Sie in der integrierten Entwicklungsumgebung (IDE), die Sie für die Entwicklungsumgebung auswählen, an dem Code



arbeiten können. Sie können ein Quell-Repository auf Ihrem lokalen Computer klonen und Änderungen zwischen Ihrem lokalen Repository und dem Remote-Repository abrufen und per Push übertragen. CodeCatalyst Sie können auch mit Quell-Repositorys arbeiten, indem Sie den Zugriff darauf in Ihrer bevorzugten IDE konfigurieren, sofern diese IDE die Verwaltung von Anmeldeinformationen unterstützt.

Repository-Namen müssen innerhalb eines CodeCatalyst Projekts eindeutig sein.

Themen

- [Erstellen eines Quell-Repositorys](#)
- [Ein Quell-Repository verlinken](#)
- [Quell-Repository anzeigen](#)
- [Einstellungen für ein Quell-Repository bearbeiten](#)
- [Klonen eines Quell-Repositorys](#)
- [Löschen eines Quell-Repositorys](#)

## Erstellen eines Quell-Repositorys

Wenn Sie ein Projekt mithilfe eines Blueprints in Amazon erstellen CodeCatalyst, CodeCatalyst erstellt ein Quell-Repository für Sie. Dieses Quell-Repository enthält zusätzlich zu den Konfigurationsinformationen für die Workflows und andere Ressourcen, die für Sie erstellt wurden, Beispielcode. Dies ist die empfohlene Methode, um mit Repositorys in CodeCatalyst zu beginnen. Sie können wählen, ob Sie Repositorys für ein Projekt erstellen möchten. Diese Repositorys enthalten eine einzige Datei, eine README.md-Datei, die Sie jederzeit bearbeiten oder löschen können. Abhängig von Ihren Optionen beim Erstellen eines Quell-Repositorys können Repositorys auch eine Datei enthalten. `.gitignore`

Um ein Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt.
3. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.
4. Wählen Sie Repository hinzufügen und anschließend Repository erstellen aus.
5. Geben Sie im Feld Repository-Name einen Namen für das Repository ein. In diesem Handbuch verwenden wir einen anderen Namen *codecatalyst-source-repository*, aber Sie können

einen anderen Namen wählen. Repository-Namen müssen innerhalb eines Projekts eindeutig sein. Weitere Hinweise zu den Anforderungen für Repository-Namen finden Sie unter [Kontingente für Quell-Repositorys in CodeCatalyst](#).

6. (Optional) Fügen Sie unter Beschreibung eine Beschreibung für das Repository hinzu, damit andere Benutzer im Projekt verstehen, wofür das Repository verwendet wird.
7. (Optional) Fügen Sie eine `.gitignore` Datei für den Codetyp hinzu, den Sie pushen möchten.
8. Wählen Sie Erstellen.

#### Note

CodeCatalyst fügt Ihrem Repository eine README .md Datei hinzu, wenn Sie sie erstellen. CodeCatalyst erstellt außerdem einen ersten Commit für das Repository in einem Standard-Branch namens main. Sie können die Datei README.md bearbeiten oder löschen, aber Sie können den Standard-Branch nicht ändern oder löschen.

## Ein Quell-Repository verlinken

Wenn Sie ein Quell-Repository mit einem Projekt verknüpfen, können Sie Repositorys einbeziehen, die über eine CodeCatalyst Erweiterung für den Dienst verfügen, der das Repository hostet, sofern diese Erweiterung für Ihren Bereich installiert ist. Nur Benutzer mit der Rolle Space-Administrator können Erweiterungen installieren. Sobald die Erweiterung installiert ist, können Sie Links zu Repositorys herstellen, die für den Zugriff durch diese Erweiterung konfiguriert sind. Weitere Informationen finden Sie unter [Erweiterungen installieren und deinstallieren in CodeCatalyst](#) oder folgen Sie [Schnellstart: GitHub Repositorys verwenden in CodeCatalyst](#).

#### Note

Sie können ein Repository nur mit einem Projekt in einem Bereich verknüpfen. Sie können ein archiviertes Repository nicht verknüpfen. Sie können zwar ein leeres Repository verknüpfen, aber Sie können es CodeCatalyst erst verwenden, wenn Sie es mit einem ersten Commit initialisiert haben, das einen Standard-Branch erstellt.

### Um ein Quell-Repository zu verlinken

1. Navigieren Sie zu dem Projekt, mit dem Sie ein Repository verknüpfen möchten.

**Note**

Bevor Sie ein Repository verknüpfen können, muss ein Benutzer mit der Space-Administratorrolle zuerst die Erweiterung für den Anbieter installieren, der das Repository hostet. Weitere Informationen finden Sie unter [Eine Erweiterung installieren](#).

2. Wählen Sie im Navigationsbereich Code und anschließend Quell-Repositories aus.
3. Wählen Sie Repository hinzufügen und anschließend Repository verknüpfen aus.
4. Wählen Sie unter Repository-Anbieter den Namen des Anbieters aus. Wählen Sie GitHub unter Konto das verknüpfte Konto aus, das Sie für die Arbeit im Repository verwenden möchten, und wählen Sie dann unter Repository den Namen des Repositories aus.

**Note**

Sie können nur ein aktives Repository verknüpfen. Sie können ein archiviertes Repository nicht verknüpfen.

5. Überprüfen Sie die Informationen und wählen Sie dann Link aus.

**Note**

Sie können ein Repository nur mit einem Projekt in einem Space verknüpfen.

Um die Verknüpfung eines Repositories mit einem Projekt aufzuheben, benötigen Sie die Rolle eines Space-Administrators. Weitere Informationen finden Sie unter [Verwaltung von GitHub Repositorien in CodeCatalyst](#).

## Quell-Repository anzeigen

Sie können die mit einem Projekt verknüpften Quell-Repositories in Amazon CodeCatalyst einsehen. Für Quell-Repositories in CodeCatalyst bietet die Übersichtsseite für ein Repository einen schnellen Überblick über Informationen und Aktivitäten in diesem Repository, einschließlich:

- Die Beschreibung des Repositories, falls vorhanden
- Die Anzahl der Branches im Repository
- Die Anzahl der offenen Pull-Requests für das Repository

- Die Anzahl verwandter Workflows für das Repository
- Die Dateien und Ordner in der Standardverzweigung oder in der Verzweigung, die Sie wählen
- Der Titel, der Autor und das Datum der letzten Übertragung in den angezeigten Zweig
- Der Inhalt der README.md-Datei wurde in Markdown gerendert, sofern eine README.md-Datei enthalten ist

Diese Seite bietet auch Links zu den Commits, Branches und Pull-Requests für das Repository sowie eine schnelle Möglichkeit, einzelne Dateien zu öffnen, anzusehen und zu bearbeiten.

#### Note

Sie können diese Informationen zu verknüpften Repositories nicht in der CodeCatalyst Konsole einsehen. Um Informationen über verknüpfte Repositories anzuzeigen, wählen Sie den Link in der Liste der Repositories, um das Repository in dem Dienst zu öffnen, der es hostet.

Um zu den Quell-Repositories für ein Projekt zu navigieren

1. Navigieren Sie zu Ihrem Projekt und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie auf der Übersichtsseite für Ihr Projekt das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen.
  - Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus. Wählen Sie unter Quell-Repositories den Namen des Repositories aus der Liste aus. Sie können die Liste der Repositories filtern, indem Sie einen Teil des Repository-Namens in die Filterleiste eingeben.
2. Sehen Sie sich auf der Startseite des Repositories den Inhalt des Repositories und Informationen zu den zugehörigen Ressourcen an, z. B. die Anzahl der Pull-Requests und Workflows. Standardmäßig werden die Inhalte für den Standard-Branch angezeigt. Sie können die Ansicht ändern, indem Sie einen anderen Zweig aus der Dropdownliste auswählen.

#### Tip

Sie können auch schnell zu den Repositories Ihres Projekts navigieren, indem Sie auf der Seite mit der Projektzusammenfassung die Option Projektcode anzeigen auswählen.

## Einstellungen für ein Quell-Repository bearbeiten

Du kannst die Einstellungen für dein Repository verwalten, einschließlich der Bearbeitung der Beschreibung eines Repositorys, der Auswahl des Standard-Branche, der Erstellung und Verwaltung von Branch-Regeln und der Erstellung und Verwaltung von Genehmigungsregeln für Pull Requests in CodeCatalyst. Auf diese Weise können die Projektmitglieder verstehen, wofür das Repository verwendet wird, und Sie können die vom Team verwendeten bewährten Methoden und Prozesse durchsetzen.

### Note

Sie können den Namen eines Quell-Repositorys nicht bearbeiten.

Sie können den Namen, die Beschreibung oder andere Informationen für ein verknüpftes Repository nicht bearbeiten in CodeCatalyst. Um Informationen über ein verknüpftes Repository zu ändern, müssen Sie sie in dem Anbieter bearbeiten, der das verknüpfte Repository hostet. Weitere Informationen finden Sie in der Dokumentation des Dienstes, der das verknüpfte Repository hostet.

Um die Einstellungen eines Repositorys zu bearbeiten

1. Navigieren Sie in der CodeCatalyst Konsole zu dem Projekt, das das Quell-Repository enthält, dessen Einstellungen Sie bearbeiten möchten.
2. Wählen Sie auf der Übersichtsseite Ihres Projekts das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus.
3. Wählen Sie auf der Übersichtsseite für das Repository die Option Mehr und anschließend Einstellungen verwalten aus.
4. Führen Sie eine oder mehrere der folgenden Aktionen aus:
  - Bearbeiten Sie die Beschreibung des Repositorys und wählen Sie dann Speichern.
  - Um den Standardzweig für das Repository zu ändern, wählen Sie unter Standardzweig die Option Bearbeiten aus. Weitere Informationen finden Sie unter [Den Standardzweig für ein Repository anzeigen und ändern](#).
  - Um eine Regel hinzuzufügen, zu entfernen oder zu ändern, nach der Projektrollen berechtigt sind, bestimmte Aktionen in einem Branch auszuführen, zu entfernen oder zu ändern, wählen

Sie unter Branch-Regeln die Option Bearbeiten aus. Weitere Informationen finden Sie unter [Zulässige Aktionen für einen Branch mit Branch-Regeln verwalten](#).

- Um eine Genehmigungsregel für das Zusammenführen von Pull-Requests zu einem Branch hinzuzufügen, zu entfernen oder zu ändern, wählen Sie unter Genehmigungsregeln die Option Bearbeiten aus. Weitere Informationen finden Sie unter [Anforderungen für das Zusammenführen einer Pull-Anfrage mit Genehmigungsregeln verwalten](#).

## Klonen eines Quell-Repositorys

Um effektiv mit mehreren Dateien, Branches und Commits in Quell-Repositorys zu arbeiten, klonen Sie das Quell-Repository auf Ihren lokalen Computer und verwenden Sie einen Git-Client oder eine integrierte Entwicklungsumgebung (IDE), um Änderungen vorzunehmen. Übernehmen Sie Ihre Änderungen und übertragen Sie sie in das Quell-Repository, um mit CodeCatalyst Funktionen wie Issues und Pull-Requests arbeiten zu können. Sie können sich auch dafür entscheiden, eine Entwicklungsumgebung zu erstellen, um an Code zu arbeiten. Beim Erstellen einer Entwicklungsumgebung werden das von Ihnen angegebene Repository und die von Ihnen angegebene Verzweigung automatisch in die Entwicklungsumgebung geklont.

### Note

Sie können verknüpfte Repositorys nicht in der CodeCatalyst Konsole klonen oder Entwicklungsumgebungen für sie erstellen. Um ein verknüpft Repository lokal zu klonen, wählen Sie den Link in der Liste der Repositorys aus, um das Repository in dem Dienst zu öffnen, der es hostet, und klonen Sie es dann. Weitere Informationen finden Sie in der Dokumentation des Dienstes, der das verknüpfte Repository hostet.

Um eine Entwicklungsumgebung aus einem Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich Code und anschließend Source Repositories aus.
3. Wählen Sie das Quell-Repository aus, in dem Sie am Code arbeiten möchten.
4. Wählen Sie Create Dev Environment aus.
5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
6. Führen Sie eine der folgenden Aktionen aus:

- Wählen Sie In vorhandenem Zweig arbeiten und wählen Sie dann im Dropdownmenü Bestehender Zweig einen Zweig aus.
  - Wählen Sie In neuer Filiale arbeiten aus, geben Sie einen Filialnamen in das Feld Filialname ein und wählen Sie aus dem Drop-down-Menü Filiale erstellen aus eine Filiale aus, aus der die neue Filiale erstellt werden soll.
7. Fügen Sie optional einen Namen für die Entwicklungsumgebung hinzu oder bearbeiten Sie deren Konfiguration.
  8. Wählen Sie Erstellen.

### Um ein Quell-Repository zu klonen

1. Navigiere zu deinem Projekt.
2. Wählen Sie auf der Übersichtsseite Ihres Projekts das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können die Liste der Repositorys filtern, indem Sie einen Teil des Repository-Namens in die Filterleiste eingeben.
- 3.
4. Wählen Sie Repository klonen. Kopieren Sie die Klon-URL für das Repository.


#### Note

Wenn Sie kein Personal Access Token (PAT) haben, wählen Sie Create token. Kopieren Sie das Token und speichern Sie es an einem sicheren Ort. Sie verwenden diese PAT, wenn Sie von Ihrem Git-Client oder Ihrer integrierten Entwicklungsumgebung (IDE) zur Eingabe eines Kennworts aufgefordert werden.

5. Führen Sie eine der folgenden Aktionen aus:
  - Um ein Repository auf Ihren lokalen Computer zu klonen, öffnen Sie ein Terminal oder eine Befehlszeile und führen Sie den git clone Befehl mit der Klon-URL nach dem Befehl aus. Beispielsweise:

```
git clone https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Wenn Sie zur Eingabe eines Kennworts aufgefordert werden, fügen Sie die zuvor gespeicherte PAT ein.


 Note

Wenn Ihr Betriebssystem die Verwaltung von Anmeldeinformationen unterstützt oder Sie ein Anmeldeinformationsverwaltungssystem installiert haben, müssen Sie die PAT nur einmal angeben. Wenn nicht, müssen Sie möglicherweise die PAT für jede Git-Operation angeben. Als bewährte Methode sollten Sie sicherstellen, dass Ihr Anmeldeinformationsverwaltungssystem Ihre PAT sicher speichert. Nehmen Sie die PAT nicht als Teil der Klon-URL-Zeichenfolge auf.

- Um ein Repository mit einer IDE zu klonen, folgen Sie der Dokumentation für Ihre IDE. Wählen Sie die Option zum Klonen eines Git-Repositorys und geben Sie die URL an. Wenn Sie zur Eingabe eines Passworts aufgefordert werden, geben Sie das PAT an.

## Löschen eines Quell-Repositorys

Wenn ein Quell-Repository für ein CodeCatalyst Amazon-Projekt nicht mehr benötigt wird, können Sie es löschen. Durch das Löschen eines Quell-Repositorys werden auch alle im Repository gespeicherten Projektinformationen gelöscht. Wenn Workflows vom Quell-Repository abhängen, werden diese Workflows nach dem Löschen des Repositorys aus der Liste der Projekt-Workflows gelöscht. Probleme, die auf das Quell-Repository verweisen, werden nicht gelöscht oder geändert, aber alle Links zum Quell-Repository, die zu Issues hinzugefügt wurden, schlagen fehl, sobald das Repository gelöscht wird.

 Important

Das Löschen eines Quell-Repositorys kann nicht rückgängig gemacht werden. Nachdem Sie ein Quell-Repository gelöscht haben, können Sie es nicht mehr klonen, Daten daraus abrufen oder Daten dorthin übertragen. Durch das Löschen eines Quell-Repositorys werden keine lokalen Kopien dieses Repositorys (lokale Repos) gelöscht. Verwenden Sie die Verzeichnis- und Dateiverwaltungstools Ihres lokalen Computers, um ein lokales Repository zu löschen.



**Note**

Sie können ein verknüpftes Repository nicht in der CodeCatalyst Konsole löschen. Um ein verknüpftes Repository zu löschen, wählen Sie den Link in der Liste der Repositories aus, um das Repository in dem Dienst zu öffnen, der es hostet, und löschen Sie es dann. Weitere Informationen finden Sie in der Dokumentation des Dienstes, der das verknüpfte Repository hostet.

Informationen zum Entfernen eines verknüpften Repositories aus einem Projekt finden Sie unter [Verwaltung von GitHub Repositorien in CodeCatalyst](#).

### Um ein Quell-Repository zu löschen

1. Navigieren Sie zu dem Projekt, das das Quell-Repository enthält, das Sie löschen möchten.
2. Wählen Sie auf der Übersichtsseite für Ihr Projekt das gewünschte Repository aus der Liste aus und klicken Sie dann auf Repository anzeigen. Sie können auch im Navigationsbereich Code und dann Quell-Repositories auswählen. Wählen Sie den Namen des Repositories aus der Liste der Quell-Repositories für das Projekt aus.
3. Wählen Sie auf der Startseite des Repositories die Option Mehr und anschließend Repository löschen aus.
4. Überprüfe den Branch, den Pull Request und die zugehörigen Workflow-Informationen, um sicherzustellen, dass du kein Repository löschst, das noch verwendet wird oder dessen Arbeit noch nicht abgeschlossen ist. Wenn Sie fortfahren möchten, geben Sie Löschen ein und wählen Sie dann Löschen aus.

## Zusammenarbeit mit Filialen bei Amazon CodeCatalyst

In Git sind Branches Zeiger oder Verweise auf einen Commit. In der Entwicklung sind sie eine praktische Möglichkeit, die Arbeit zu organisieren. Du kannst Branches verwenden, um die Arbeit an einer neuen oder anderen Version von Dateien voneinander zu trennen, ohne die Arbeit in anderen Branches zu beeinträchtigen. Sie können Verzweigungen verwenden, um neue Funktionen zu entwickeln, eine bestimmte Version Ihres Projekts zu speichern und vieles mehr. Sie können Regeln für Branches in Quell-Repositories konfigurieren, um bestimmte Aktionen in einem Branch auf bestimmte Rollen in diesem Projekt zu beschränken.

Quell-Repositorys in Amazon CodeCatalyst haben Inhalte und einen Standardzweig, unabhängig davon, wie Sie sie erstellen. Verknüpfte Repositorys haben möglicherweise keinen Standard-Branch oder Inhalt, sind aber CodeCatalyst erst nutzbar, wenn Sie sie initialisieren und einen Standard-Branch erstellen. Wenn Sie ein Projekt mithilfe eines Blueprints erstellen, CodeCatalyst wird ein Quell-Repository für dieses Projekt erstellt, das eine README.md-Datei, Beispielcode, Workflow-Definitionen und andere Ressourcen enthält. Wenn Sie ein Quell-Repository erstellen, ohne einen Blueprint zu verwenden, wird Ihnen als erster Commit eine README.md-Datei hinzugefügt, und ein Standardzweig wird für Sie erstellt. Dieser Standardzweig heißt main. Dieser Standardzweig wird als Basis- oder Standardzweig in lokalen Repositorys (Repos) verwendet, wenn Benutzer das Repository klonen.

### Note

Sie können den Standardzweig nicht löschen. Der erste Branch, der für ein Quell-Repository erstellt wurde, ist der Standard-Branch für dieses Repository. Darüber hinaus zeigt die Suche nur Ergebnisse aus dem Standardzweig an. Sie können in anderen Branchen nicht nach Code suchen.

Beim Erstellen eines Repositorys in wird CodeCatalyst auch ein erster Commit erstellt, der einen Standard-Branch mit einer darin enthaltenen README.md-Datei erstellt. Der Name dieses Standardzweigs ist main. Dies ist der Standard-Zweigname, der in den Beispielen in diesem Handbuch verwendet wird.


## Themen

- [Einen Zweig erstellen und löschen](#)
- [Den Standardzweig für ein Repository anzeigen und ändern](#)
- [Zulässige Aktionen für einen Branch mit Branch-Regeln verwalten](#)
- [Git-Befehle für Branches](#)
- [Filialen und Details anzeigen](#)

## Einen Zweig erstellen und löschen

Sie können die CodeCatalyst Konsole verwenden, um Branches in einem CodeCatalyst Repository zu erstellen und zu löschen. Die von Ihnen erstellten Branches sind für andere Benutzer sichtbar, wenn sie das nächste Mal Änderungen aus dem Repository abrufen. Wenn Sie einen Branch

löschen, verbleiben Kopien dieses Branches in den Klonen des Repositorys auf lokalen Computern, bis Benutzer diese Änderungen abrufen und synchronisieren.

 Tip

Sie können im Rahmen der Erstellung einer Entwicklungsumgebung auch Branches erstellen, um an Ihrem Code zu arbeiten. Weitere Informationen finden Sie unter [Erstellen einer Entwicklungsumgebung](#).

Du kannst Git auch verwenden, um Branches zu erstellen und zu löschen. Weitere Informationen finden Sie unter [Allgemeine Git-Befehle für Branches](#).

Um einen Zweig (Konsole) zu erstellen

1. Navigieren Sie in der CodeCatalyst Konsole zu dem Projekt, in dem sich Ihr Quell-Repository befindet.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.
3. Wählen Sie das Repository aus, in dem Sie einen Branch erstellen möchten.
4. Wählen Sie auf der Übersichtsseite des Repositorys die Option Mehr und dann Zweig erstellen aus.
5. Geben Sie einen Namen für den Branch ein.
6. Wählen Sie einen Zweig aus, aus dem der Zweig erstellt werden soll, und klicken Sie dann auf Erstellen.

Um einen Zweig zu löschen (Konsole)

1. Navigieren Sie zu dem Projekt, in dem sich Ihr Repository befindet.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.

Wählen Sie das Repository aus, in dem Sie einen Branch löschen möchten.

3. Wählen Sie auf der Übersichtsseite des Repositorys die Dropdownauswahl neben dem Branch-Namen und wählen Sie dann Alle anzeigen aus.
4. Wählen Sie den Zweig aus, den Sie löschen möchten, und klicken Sie dann auf Zweig löschen.

**Note**

Sie können den Standardzweig für ein Repository nicht löschen.

5. Ein Bestätigungsdiaologfeld wird angezeigt. Es zeigt das Repository, die Anzahl der geöffneten Pull-Requests und die Anzahl der Workflows, die dem Branch zugeordnet sind.
6. Um das Löschen des Branches zu bestätigen, geben Sie Löschen in das Textfeld ein und wählen Sie dann Löschen aus.

## Den Standardzweig für ein Repository anzeigen und ändern

Sie können angeben, welcher Branch als Standard-Branch in einem Quell-Repository in Amazon verwendet werden soll CodeCatalyst. Alle Quell-Repositorys in CodeCatalyst haben Inhalte und einen Standardzweig, unabhängig davon, wie Sie sie erstellen. Wenn Sie einen Blueprint verwenden, um ein Projekt zu erstellen, heißt der Standardzweig im Quell-Repository, der für dieses Projekt erstellt wurde, main. Der Inhalt des Standardzweigs wird automatisch auf der Übersichtsseite für dieses Repository angezeigt.

Der Standard-Branch wird etwas anders behandelt als alle anderen Branches in einem Quell-Repository. Neben seinem Namen steht ein spezielles Label, Default. Der Standard-Branch wird als Basis- oder Standard-Branch in lokalen Repositorys (Repos) verwendet, wenn Benutzer das Repository mit einem Git-Client auf lokale Computer klonen. Er ist auch der Standard, der bei der Erstellung von Workflows zum Speichern von Workflow-YAML-Dateien und zum Speichern von Informationen für Probleme verwendet wird. Wenn Sie „Suchen in“ verwenden CodeCatalyst, wird nur der Standardzweig eines Repositorys durchsucht. Da der Standardzweig für so viele Aspekte von Projekten von grundlegender Bedeutung ist, können Sie einen Branch nicht löschen, wenn er als Standard-Branch angegeben ist. Sie können sich jedoch dafür entscheiden, einen anderen Zweig als Standardzweig zu verwenden. Wenn Sie dies tun, werden alle [Verzweigungsregeln](#), die auf den früheren Standardzweig angewendet wurden, automatisch auf den Zweig angewendet, den Sie als Standardzweig angeben.

**Note**

Sie benötigen die Rolle eines Projektadministrators, um den Standardzweig für Quell-Repositorys in CodeCatalyst Projekten ändern zu können. Dies gilt nicht für verknüpfte Repositorys.

## Um den Standard-Branch für ein Repository anzuzeigen und zu ändern

1. Navigieren Sie zu dem Projekt, in dem sich Ihr Repository befindet.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositories auswählen.

Wählen Sie das Repository aus, in dem Sie die Einstellungen, einschließlich des Standardzweigs, anzeigen möchten.

3. Wählen Sie auf der Übersichtsseite des Repositorys die Option Mehr und anschließend Einstellungen verwalten aus.
4. Im Standardzweig wird der Name des als Standardverzweigung angegebenen Zweigs zusammen mit der Bezeichnung Standard neben dem Namen angezeigt. Dieselbe Bezeichnung wird neben dem Zweignamen in der Liste der Zweige unter Zweige angezeigt.
5. Um den Standardzweig zu ändern, wählen Sie Bearbeiten.

### Note

Sie müssen im Projekt über die Rolle des Projektadministrators verfügen, um den Standardzweig ändern zu können.

6. Wählen Sie aus der Dropdownliste den Namen des Zweigs aus, den Sie zum Standardzweig machen möchten, und wählen Sie dann Speichern aus.

## Zulässige Aktionen für einen Branch mit Branch-Regeln verwalten

Wenn Sie einen Branch erstellen, sind je nach den Berechtigungen für diese Rolle bestimmte Aktionen für diesen Branch zulässig. Sie können ändern, welche Aktionen für einen bestimmten Zweig zulässig sind, indem Sie Zweigregeln konfigurieren. Branch-Regeln basieren auf der Rolle, die ein Benutzer in Ihrem Projekt innehat. Sie können sich dafür entscheiden, einige vordefinierte Aktionen, wie das Übertragen von Commits an einen Branch, auf Benutzer mit einer bestimmten Rolle in einem Projekt zu beschränken. Auf diese Weise können Sie bestimmte Branches in einem Projekt schützen, indem Sie einschränken, welche Rollen bestimmte Aktionen ausführen dürfen. Wenn Sie beispielsweise eine Verzweigungsregel so konfigurieren, dass nur Benutzer mit der Rolle des Projektadministrators diesen Zweig zusammenführen oder zu diesem Zweig wechseln können, können Benutzer mit anderen Rollen im Projekt keine Änderungen am Code in diesem Zweig vornehmen.

Sie sollten alle Auswirkungen der Erstellung einer Regel für einen Zweig sorgfältig abwägen. Wenn Sie sich beispielsweise dafür entscheiden, Pushs auf einen Branch auf Benutzer mit der Rolle Project Administrator zu beschränken, können Benutzer mit der Rolle Contributor keine Workflows in diesem Branch erstellen oder bearbeiten, da das Workflow-YAML in diesem Branch gespeichert ist und diese Benutzer keine Änderungen an der YAML festschreiben und per Push übertragen können. Es hat sich bewährt, alle Branch-Regeln zu testen, nachdem Sie sie erstellt haben, um sicherzustellen, dass sie keine Auswirkungen haben, die Sie nicht beabsichtigt haben. Sie können Branch-Regeln auch in Verbindung mit Genehmigungsregeln für Pull Requests verwenden. Weitere Informationen finden Sie unter [Anforderungen für das Zusammenführen einer Pull-Anfrage mit Genehmigungsregeln verwalten](#).

#### Note

Sie benötigen die Rolle eines Projektadministrators, um Branch-Regeln für Quell-Repositorys in CodeCatalyst Projekten verwalten zu können. Sie können keine Branch-Regeln für verknüpfte Repositorys erstellen.

Sie können nur Branch-Regeln erstellen, die restriktiver sind als die Standardberechtigungen für die Rolle. Sie können keine Verzweigungsregeln erstellen, die freizügiger sind, als es die Rolle eines Benutzers im Projekt zulässt. Sie können beispielsweise keine Branch-Regel erstellen, die es Benutzern mit der Rolle Reviewer ermöglicht, Push in den Branch zu übertragen.

Branch-Regeln, die auf den Standard-Branch Ihres Quell-Repositorys angewendet werden, verhalten sich etwas anders als Branch-Regeln, die auf andere Branches angewendet werden. Jede Regel, die auf den Standard-Branch angewendet wird, wird automatisch auf jeden Branch angewendet, den Sie als Standard-Branch angeben. Der Zweig, der zuvor als Standardzweig festgelegt wurde, behält weiterhin die Regeln bei, die auf ihn angewendet wurden, mit der Ausnahme, dass er nicht mehr vor dem Löschen geschützt ist. Dieser Schutz wird nur auf den aktuellen Standardzweig angewendet.

Zweigregeln haben zwei Status: Standard und Benutzerdefiniert. Standard gibt an, dass die in einem Branch zulässigen Aktionen den Berechtigungen entsprechen, die der Benutzer für Branch-Aktionen in CodeCatalyst der Rolle hat, über die der Benutzer verfügt. Weitere Informationen darüber, welche Rollen welche Berechtigungen haben, finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#). Benutzerdefiniert gibt an, dass eine oder mehrere Branch-Aktionen Aktionen mit einer bestimmten Liste von Rollen haben, die diese Aktion ausführen dürfen. Diese Liste unterscheidet sich von den Standardberechtigungen, die durch die Rolle eines Benutzers im Projekt gewährt werden.

**Note**

Wenn Sie eine Verzweigungsregel erstellen, um eine oder mehrere Aktionen für einen Zweig einzuschränken, wird die Aktion Verzweigung löschen automatisch so eingestellt, dass nur Benutzer mit der Projektadministratorrolle diesen Zweig löschen können.

In der folgenden Tabelle sind die Aktionen und die Standardeinstellungen für Rollen aufgeführt, die diese Aktionen in einem Zweig ausführen dürfen.

## Aktionen und Rollen in Zweigen

Maßnahmen in der Branche	Rollen, die diese Aktion ausführen dürfen, wenn keine Branch-Regeln angewendet werden
Mit dem Branch zusammenführen (dazu gehört auch das Zusammenführen einer Pull-Anfrage mit dem Branch)	Projektadministrator, Mitwirkender
Gehen Sie zur Filiale	Projektadministrator, Mitwirkender
Löschen Sie den Zweig	Projektadministrator, Mitwirkender
Löschen Sie den Zweig (Standardzweig)	Nicht zulässig

Sie können Branch-Regeln nicht löschen, aber Sie können sie so aktualisieren, dass Aktionen von allen Rollen zugelassen werden, die diese Aktion in einem Branch ausführen dürfen, wodurch die Regel effektiv entfernt wird.

**Note**

Sie benötigen die Rolle eines Projektadministrators, um Verzweigungsregeln für Quell-Repositorys in CodeCatalyst Projekten konfigurieren zu können. Dies gilt nicht für verknüpfte Repositorys. Verknüpfte Repositorys unterstützen die Branch-Regeln in nicht. CodeCatalyst

## Um Branch-Regeln für ein Repository anzuzeigen und zu bearbeiten

1. Navigieren Sie zu dem Projekt, in dem sich Ihr Repository befindet.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.

Wählen Sie das Repository aus, in dem Sie Branch-Regeln anzeigen möchten.

3. Wählen Sie auf der Übersichtsseite des Repositorys die Option Branches aus.
4. Sehen Sie sich in der Spalte Branch-Regeln den Status der Regeln für jeden Branch des Repositorys an. Standard gibt an, dass die Regeln für Branch-Aktionen die Standardregeln für alle Branches sind, die in einem Quell-Repository erstellt wurden, und dass sie den Berechtigungen entsprechen, die diesen Rollen in einem Projekt gewährt wurden. Benutzerdefiniert gibt an, dass für eine oder mehrere Branch-Aktionen Regeln gelten, die eine oder mehrere Aktionen, die für diesen Branch zulässig sind, auf eine andere Gruppe von Rollen beschränken.

Um die Einzelheiten der Branch-Regeln für einen Branch einzusehen, wählen Sie neben dem Zweig, den Sie überprüfen möchten, das Wort Standard oder Benutzerdefiniert aus.

5. Um eine Zweigregel zu erstellen oder zu ändern, wählen Sie Einstellungen verwalten. Wählen Sie auf der Einstellungsseite für das Quell-Repository unter Branch-Regeln die Option Bearbeiten aus.
6. Wählen Sie unter Branch den Namen des Branches, für den Sie eine Regel konfigurieren möchten, aus der Dropdownliste aus. Wählen Sie für jeden der zulässigen Aktionstypen aus der Dropdownliste die Rollen aus, denen Sie die Aktion zuweisen möchten, und wählen Sie dann Speichern aus.

## Git-Befehle für Branches

Du kannst Git verwenden, um Branches im Clone des Quell-Repositorys, das du auf deinem Computer (deinem lokalen Repo) oder in deinen Entwicklungsumgebungen hast, zu erstellen, zu verwalten und zu löschen. Anschließend kannst du deine Änderungen in dein CodeCatalyst Quell-Repository (das Remote-Repository) übertragen und per Push übertragen. Beispielsweise:



## Allgemeine Git-Befehle für Branches

Listet alle Branches im lokalen Repository auf, wobei neben deinem aktuellen Branch ein Sternchen (*) angezeigt wird.	<code>git branch</code>
Ruft Informationen über alle vorhandenen Branches im Remote-Repository in das lokale Repository ab.	<code>git fetch</code>
Listet alle Branches im lokalen Repo und alle Remote-Tracking-Zweige im lokalen Repo auf.	<code>git branch -a</code>
Listet nur Remote-Tracking-Zweige im lokalen Repo auf.	<code>git branch -r</code>
Erstellt einen Zweig im lokalen Repository unter Verwendung des angegebenen Zweignamens. Dieser Branch wird erst im Remote-Repository angezeigt, wenn Sie die Änderung festschreiben und übertragen.	<code>git branch <i>branch-name</i></code>
Erstellt einen Branch im lokalen Repository unter Verwendung des angegebenen Branch-Namens und wechselt dann zu diesem.	<code>git checkout -b <i>branch-name</i></code>
Wechselt unter Verwendung des angegebenen Zweignamens zu einem anderen Zweig im lokalen Repository.	<code>git checkout <i>other-branch-name</i></code>
Verschiebt einen Branch vom lokalen Repository zum Remote-Repository und verwendet dabei den im lokalen Repository angegebenen Spitznamen für das Remote-Repository und den angegebenen Branch-Namen. Richtet außerdem Upstream-Tracking-Informationen für den Branch im lokalen Repository ein.	<code>git push -u <i>remote-name</i> <i>branch-name</i></code>

Führt Änderungen von einem anderen Zweig im lokalen Repository mit dem aktuellen Zweig im lokalen Repository zusammen.

```
git merge from-other-branch-name
```

Löscht einen Zweig im lokalen Repository, es sei denn, er enthält Werke, die noch nicht zusammengeführt wurden.

```
git branch -d branch-name
```

Löscht einen Branch im Remote-Repository mit dem angegebenen Spitznamen, den das lokale Repository für das Remote-Repository hat, und dem angegebenen Branch-Namen. (Beachten Sie die Verwendung des Doppelpunkts (:).) Sie können dies auch `--delete` als Teil des Befehls angeben.

```
git push remote-name :branch-name
```

```
git push remote-name --delete  
branch-name
```

Weitere Informationen findest du in deiner Git-Dokumentation.

## Filialen und Details anzeigen

In der CodeCatalyst Amazon-Konsole können Sie Informationen zu Remote-Branches in Amazon einsehen CodeCatalyst, einschließlich Einzelheiten zu den Dateien, Ordnern und dem letzten Commit für eine bestimmte Filiale. Sie können auch Git-Befehle und Ihr lokales Betriebssystem verwenden, um diese Informationen für entfernte und lokale Niederlassungen anzuzeigen.

Um Branches anzuzeigen (Konsole)

1. Navigieren Sie in der CodeCatalyst Konsole zu dem Projekt, das das Quell-Repository enthält, in dem Sie die Branches anzeigen möchten. Wählen Sie Code, dann Quell-Repositories und anschließend das Quell-Repository aus.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositories für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositories auswählen.

Wählen Sie das Repository aus, in dem Sie einen Branch anzeigen möchten.

3. Der Standardzweig des Repositorys wird angezeigt. Sie können eine Liste der Dateien und Ordner im Branch, Informationen über den letzten Commit und den Inhalt der README.md-

Datei, falls sie im Branch vorhanden ist, sehen. Um die Informationen für einen anderen Branch anzuzeigen, wählen Sie ihn aus der Dropdownliste der Branches für das Repository aus.

- Um alle Branches für ein Repository anzuzeigen, wählen Sie Alle anzeigen. Auf der Seite „Branches“ werden Informationen zum Namen, zum letzten Commit und zu den Regeln für jeden Branch angezeigt.

Informationen dazu, wie du Git und dein Betriebssystem zum Anzeigen von Branches und Details verwendest, findest du unter [Allgemeine Git-Befehle für Branches](#), in deiner Git-Dokumentation und in deiner Betriebssystemdokumentation.

## Arbeiten mit Dateien in Amazon CodeCatalyst

Bei Amazon CodeCatalyst ist eine Datei eine versionsgesteuerte, eigenständige Information, die Ihnen und anderen Benutzern des Quell-Repositorys und Zweigs, in dem die Datei gespeichert ist, zur Verfügung steht. Sie können Ihre Repository-Dateien mit einer Verzeichnisstruktur organisieren. CodeCatalyst verfolgt automatisch jede festgeschriebene Änderung an einer Datei. Sie können verschiedene Versionen einer Datei in verschiedenen Repository-Zweigen speichern.

Um mehrere Dateien in einem Quell-Repository hinzuzufügen oder zu bearbeiten, können Sie einen Git-Client, eine Entwicklungsumgebung oder eine integrierte Entwicklungsumgebung (IDE) verwenden. Um eine einzelne Datei hinzuzufügen oder zu bearbeiten, können Sie die CodeCatalyst Konsole verwenden.

### Themen

- [Eine Datei erstellen oder hinzufügen](#)
- [Eine Datei ansehen](#)
- [Bearbeiten einer Datei](#)
- [Eine Datei umbenennen oder löschen](#)

## Eine Datei erstellen oder hinzufügen

Um Dateien zu erstellen und einem Quell-Repository hinzuzufügen, können Sie die CodeCatalyst Amazon-Konsole, eine Entwicklungsumgebung, eine verbundene integrierte Entwicklungsumgebung (IDE) oder einen Git-Client verwenden. Die CodeCatalyst Konsole enthält einen Code-Editor zum Erstellen von Dateien. Dieser Editor ist eine bequeme Möglichkeit, eine einfache Datei, z. B. eine

README.md-Datei, in einem Zweig eines Repositorys zu erstellen oder zu bearbeiten. Wenn Sie an mehr als einer Datei arbeiten, sollten Sie erwägen, eine Entwicklungsumgebung [zu erstellen](#).

Um eine Entwicklungsumgebung aus einem Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich Code und anschließend Source Repositories aus.
3. Wählen Sie das Quell-Repository aus, in dem Sie am Code arbeiten möchten.
4. Wählen Sie Create Dev Environment aus.
5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
6. Führen Sie eine der folgenden Aktionen aus:
  - Wählen Sie In vorhandenem Zweig arbeiten und wählen Sie dann im Dropdownmenü Bestehender Zweig einen Zweig aus.
  - Wählen Sie In neuer Filiale arbeiten aus, geben Sie einen Filialnamen in das Feld Filialname ein und wählen Sie aus dem Drop-down-Menü Filiale erstellen aus eine Filiale aus, aus der die neue Filiale erstellt werden soll.
7. Fügen Sie optional einen Namen für die Entwicklungsumgebung hinzu oder bearbeiten Sie deren Konfiguration.
8. Wählen Sie Erstellen.

Um eine Datei in der CodeCatalyst Konsole zu erstellen

1. Navigieren Sie zu dem Projekt, in dem Sie eine Datei erstellen möchten. Weitere Informationen zum Navigieren zu einem Repository finden Sie unter [Quell-Repository anzeigen](#).
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.

Wählen Sie das Repository aus, in dem Sie die Datei erstellen möchten.

3. (Optional) Wählen Sie den Zweig aus, in dem Sie die Datei erstellen möchten, wenn Sie die Datei in einem anderen Zweig als dem Standardzweig erstellen möchten.
4. Wählen Sie „Datei erstellen“.
5. Geben Sie im Feld Dateiname den Namen der Datei ein. Fügen Sie den Inhalt der Datei im Editor hinzu.

**i** Tip

Wenn Sie die Datei in einem Unterordner oder Unterverzeichnis im Stammverzeichnis des Zweigs erstellen möchten, fügen Sie diese Struktur als Teil des Dateinamens hinzu.

Wenn Sie mit Ihren Änderungen zufrieden sind, wählen Sie Commit.

- Überprüfen Sie unter Dateiname den Namen der Datei und nehmen Sie die gewünschten Änderungen daran vor. Wählen Sie optional aus der Liste der verfügbaren Zweige in Branch den Zweig aus, in dem Sie die Datei erstellen möchten. Geben Sie im Feld Commit-Nachricht optional eine kurze, aber informative Beschreibung ein, warum Sie diese Änderung vorgenommen haben. Dies wird als grundlegende Commit-Information für den Commit angezeigt, der die Datei zum Quell-Repository hinzufügt.
- Wählen Sie Commit, um die Datei zu übertragen und in das Quell-Repository zu übertragen.

Du kannst auch Dateien zu einem Quell-Repository hinzufügen, indem du sie auf deinen lokalen Computer klonst und einen Git-Client oder eine verbundene integrierte Entwicklungsumgebung (IDE) verwendest, um deine Dateien und Änderungen zu übertragen.

**i** Note

Wenn Sie ein Git-Submodul hinzufügen möchten, müssen Sie einen Git-Client oder eine Entwicklungsumgebung verwenden und den `git submodule add` Befehl ausführen. Sie können Git-Submodule nicht in der CodeCatalyst Konsole hinzufügen oder anzeigen oder die Unterschiede zwischen Git-Submodulen in Pull-Requests anzeigen. Weitere Informationen zu Git-Submodulen finden Sie in der [Git-Dokumentation](#).

Um eine Datei mit einem Git-Client oder einer verbundenen integrierten Entwicklungsumgebung (IDE) hinzuzufügen

- Klonen Sie Ihr Quell-Repository auf Ihren lokalen Computer. Weitere Informationen finden Sie unter [Klonen eines Quell-Repositorys](#).
- Erstellen Sie Dateien in Ihrem lokalen Repo oder kopieren Sie Dateien in Ihr lokales Repo.
- Erstelle und pushe einen Commit, indem du einen der folgenden Schritte ausführst:

- Wenn Sie einen Git-Client verwenden, führen Sie den Befehl im Terminal oder in der `git add` Befehlszeile aus und geben Sie die Namen der Dateien an, die Sie hinzufügen möchten. Um alle hinzugefügten oder geänderten Dateien hinzuzufügen, führen Sie alternativ den `git add` Befehl gefolgt von einem einzelnen oder einem doppelten Punkt aus, um anzugeben, ob Sie alle Änderungen auf der aktuellen Verzeichnisebene (einzelne Periode) oder alle Änderungen im aktuellen Verzeichnis und allen Unterverzeichnissen (doppelter Punkt) einbeziehen möchten. Um die Änderungen zu übernehmen, führen Sie den `git commit -m` Befehl aus und geben Sie eine Commit-Nachricht ein. Führen Sie den Befehl aus, um Ihre Änderungen in CodeCatalyst das Quell-Repository zu übertragen `git push`. Weitere Informationen zu Git-Befehlen finden Sie in Ihrer Git-Dokumentation und [Git-Befehle für Branches](#).
- Wenn Sie eine Entwicklungsumgebung oder eine IDE verwenden, erstellen Sie Dateien in der IDE und fügen Sie Dateien hinzu. Übergeben Sie dann Ihre Änderungen und übertragen Sie sie. Weitere Informationen finden Sie in [Entwicklungsumgebungen in CodeCatalyst](#) oder in Ihrer IDE-Dokumentation.

## Eine Datei ansehen

Sie können die Dateien in Ihrem Quell-Repository in der CodeCatalyst Amazon-Konsole anzeigen. Sie können die Dateien im Standardzweig und in allen anderen Zweigen anzeigen. Der Dateiinhalt kann je nach Zweig, den Sie anzeigen möchten, variieren.

Um Dateien in der CodeCatalyst Konsole anzuzeigen


1. Navigieren Sie zu dem Projekt, in dem Sie Dateien anzeigen möchten. Weitere Informationen finden Sie unter [Quell-Repository anzeigen](#).
2. Wählen Sie den Namen des Repositories aus der Liste der Quell-Repositories für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositories auswählen.  
  
Wählen Sie das Repository aus, in dem Sie Dateien anzeigen möchten.
3. Für den Standardzweig wird eine Liste von Dateien und Ordnern angezeigt. Dateien werden durch ein Papiersymbol gekennzeichnet, während Ordner durch ein Ordnersymbol gekennzeichnet sind.
4. Führen Sie eine der folgenden Aktionen aus:
  - Um Dateien und Ordner in einem anderen Zweig anzuzeigen, wählen Sie ihn aus der Liste der Zweige aus.

- Um einen Ordner zu erweitern, wählen Sie ihn aus der Liste aus.
5. Um den Inhalt einer bestimmten Datei anzuzeigen, wählen Sie sie aus der Liste aus. Der Inhalt der Datei wird im Zweig angezeigt. Um den Inhalt der Datei in einem anderen Zweig anzuzeigen, wählen Sie den gewünschten Zweig aus der Verzweigungsauswahl aus.

 Tip

Wenn Sie sich den Inhalt einer Datei ansehen, können Sie unter Dateien anzeigen weitere Dateien auswählen. Um eine Datei zu bearbeiten, wählen Sie Bearbeiten.

Sie können mehrere Dateien in der Konsole anzeigen. Sie können auch Dateien anzeigen, die Sie mit einem Git-Client oder einer integrierten Entwicklungsumgebung (IDE) auf Ihren lokalen Computer geklont haben. Weitere Informationen finden Sie in der Dokumentation für Ihren Git-Client oder Ihre Git-IDE.

 Note

Sie können Git-Submodule nicht in der CodeCatalyst Konsole anzeigen. Weitere Informationen zu Git-Submodulen finden Sie in der [Git-Dokumentation](#).

## Bearbeiten einer Datei

Sie können einzelne Dateien in der CodeCatalyst Amazon-Konsole bearbeiten. Um mehrere Dateien gleichzeitig zu bearbeiten, erstellen Sie entweder eine Entwicklungsumgebung oder klonen Sie das Repository und nehmen Sie Ihre Änderungen mit einem Git-Client oder einer integrierten Entwicklungsumgebung (IDE) vor. Weitere Informationen finden Sie unter [Entwicklungsumgebungen in CodeCatalyst](#) oder [Klonen eines Quell-Repositorys](#).

Um eine Datei in der CodeCatalyst Konsole zu bearbeiten

1. Navigieren Sie zu dem Projekt, in dem Sie eine Datei bearbeiten möchten. Weitere Informationen zum Navigieren zu einem Repository finden Sie unter [Quell-Repository anzeigen](#).
2. Wählen Sie das Repository aus, in dem Sie die Datei bearbeiten möchten. Wählen Sie Zweige anzeigen und wählen Sie dann den Zweig aus, in dem Sie arbeiten möchten. Wählen Sie die Datei aus der Liste der Dateien und Ordner in diesem Zweig aus.

Der Inhalt der Datei wird angezeigt.

3. Wählen Sie Bearbeiten aus.
4. Bearbeiten Sie im Editor den Inhalt der Datei und wählen Sie dann Commit. Fügen Sie optional unter Änderungen übernehmen weitere Informationen zur Änderung in der Nachricht bestätigen hinzu. Wenn Sie mit Ihren Änderungen zufrieden sind, wählen Sie Commit aus.

## Eine Datei umbenennen oder löschen

Sie können Dateien in einer Entwicklungsumgebung, lokal auf Ihrem Computer oder in einer integrierten Entwicklungsumgebung (IDE) umbenennen oder löschen. Sobald Sie die Dateien umbenannt oder gelöscht haben, übertragen Sie diese Änderungen und übertragen Sie sie in das Quell-Repository. Sie können Dateien in der CodeCatalyst Amazon-Konsole nicht umbenennen oder löschen.

## Mit Pull-Requests in Amazon arbeiten CodeCatalyst

Ein Pull Request ist die wichtigste Methode, mit der du und andere Projektmitglieder Codeänderungen von einem Branch zum anderen überprüfen, kommentieren und zusammenführen können. Sie können Pull Requests verwenden, um Codeänderungen gemeinsam auf kleinere Änderungen oder Korrekturen, wichtige Funktionserweiterungen oder neue Versionen Ihrer veröffentlichten Software zu überprüfen. Wenn du Issues verwendest, um die Arbeit an deinem Projekt nachzuverfolgen, kannst du bestimmte Issues mit Pull Requests verknüpfen, um nachzuverfolgen, welche Probleme durch die Codeänderungen im Pull Request behoben werden. Wenn du einen Pull Request erstellst, aktualisierst, kommentierst, zusammenführst oder schließt, wird automatisch eine E-Mail an den Autor des Pull Requests sowie an alle erforderlichen oder optionalen Überprüfer für den Pull Request gesendet.

### Tip

Du kannst in deinem Profil konfigurieren, über welche Pull-Request-Ereignisse du E-Mails erhältst. Weitere Informationen finden Sie unter [Benachrichtigungen in Amazon verwalten CodeCatalyst](#).

Für Pull-Requests sind zwei Branches in einem Quell-Repository erforderlich: ein Quell-Branch, der den Code enthält, den Sie überprüfen möchten, und ein Ziel-Branch, in dem Sie den überprüften



Code zusammenführen möchten. Der Quell-Branch enthält den AFTER-Commit, bei dem es sich um den Commit mit den Änderungen handelt, die Sie mit dem Ziel-Branch zusammenführen möchten. Der Ziel-Branch enthält den BEFORE-Commit, der den Status des Codes repräsentiert, bevor der Pull-Anforderungs-Branch mit dem Ziel-Branch zusammengeführt wird.

#### Note

Während Sie einen Pull-Request erstellen, ist der angezeigte Unterschied der Unterschied zwischen der Spitze des Quell-Branches und der Spitze des Ziel-Branches. Sobald du den Pull Request erstellt hast, besteht der angezeigte Unterschied zwischen der Revision des Pull-Requests, den du ausgewählt hast, und dem Commit, der bei der Erstellung des Pull-Requests der Tip des Ziel-Branches war. Weitere Informationen zu Unterschieden und Merge-Bases in Git finden Sie [git-merge-base](#) in der Git-Dokumentation.

Ein Pull-Request wird zwar für ein bestimmtes Quell-Repository und Branches erstellt, aber du kannst sie im Rahmen der Arbeit mit deinem Projekt erstellen, ansehen, überprüfen und schließen. Sie müssen das Quell-Repository nicht aufrufen, um Pull-Requests anzeigen und damit arbeiten zu können. Ein Pull-Request-Status wird bei der Erstellung auf Offen gesetzt. Der Pull-Request bleibt geöffnet, bis Sie ihn entweder in der CodeCatalyst Konsole zusammenführen, wodurch der Status in Zusammengeführt geändert wird, oder bis Sie ihn schließen, wodurch der Status in Geschlossen geändert wird.

Wenn dein Code überprüft wurde, kannst du den Status der Pull-Anfrage auf eine von mehreren Arten ändern:

- Führen Sie den Pull-Request in der CodeCatalyst Konsole zusammen. Der Code im Quellzweig der Pull-Anfrage wird mit dem Zielzweig zusammengeführt. Der Status der Pull-Anfrage wird auf Zusammengeführt geändert. Er kann nicht wieder auf „Offen“ zurückgesetzt werden.
- Führen Sie die Branches lokal zusammen, übertragen Sie Ihre Änderungen und schließen Sie dann den Pull-Request in der CodeCatalyst Konsole.
- Verwenden Sie die CodeCatalyst Konsole, um den Pull-Request ohne Zusammenführung zu schließen. Dadurch wird der Status auf Geschlossen geändert, und der Code aus dem Quell-Branch wird nicht mit dem Ziel-Branch zusammengeführt.

Bevor Sie eine Pull-Anforderung erstellen, müssen Sie folgende Schritte ausführen:

- Übergeben Sie die Codeänderungen, die Sie überprüfen möchten, und übertragen Sie sie in einen Branch (den Quell-Branch).
- Richten Sie Benachrichtigungen für Ihr Projekt ein, damit andere Benutzer über alle Workflows informiert werden können, die ausgeführt werden, wenn Sie eine Pull-Anfrage erstellen. (Dieser Schritt ist optional, wird aber empfohlen.)

## Themen

- [Erstellen einer Pull-Anforderung](#)
- [Pull-Requests anzeigen](#)
- [Anforderungen für das Zusammenführen einer Pull-Anfrage mit Genehmigungsregeln verwalten](#)
- [Einen Pull-Request überprüfen](#)
- [Aktualisierung einer Pull-Anfrage](#)
- [Eine Pull-Anfrage zusammenführen](#)
- [Einen Pull-Request schließen](#)

## Erstellen einer Pull-Anforderung

Die generativen KI-Funktionen in Amazon CodeCatalyst befinden sich in der Vorschauversion und können sich ändern. Sie sind nur in der Region USA West (Oregon) verfügbar. Der Zugang zu generativen KI-Funktionen ist je nach Stufe unterschiedlich. Weitere Informationen finden Sie unter [-Preisgestaltung](#).

Durch das Erstellen von Pull-Anforderungen können andere Benutzer Ihre Codeänderungen sehen und prüfen, bevor Sie diese mit einem anderen Branch zusammenführen. Zunächst erstellen Sie einen Branch für Ihre Codeänderungen. Dieser wird als Quell-Branch einer Pull-Anforderung bezeichnet. Nachdem Sie die Änderungen in das Repository übernommen und per Push übertragen haben, können Sie eine Pull-Anfrage erstellen, die den Inhalt des Quell-Branches mit dem Inhalt des Ziel-Branches vergleicht.

Sie können in der CodeCatalyst Amazon-Konsole einen Pull-Request von einem bestimmten Branch aus, von der Pull-Requests-Seite oder von der Projektübersicht aus erstellen. Wenn Sie einen Pull Request von einem bestimmten Branch aus erstellen, werden der Repository-Name und der Quell-Branch automatisch auf der Seite zur Erstellung von Pull-Requests angezeigt. Wenn du einen Pull-

Request erstellt, erhältst du automatisch E-Mails über alle Aktualisierungen des Pull-Requests sowie darüber, ob der Pull-Request zusammengeführt oder geschlossen wurde.

### Note

Während du einen Pull Request erstellst, ist der angezeigte Unterschied der Unterschied zwischen der Spitze des Quell-Branche und der Spitze des Ziel-Branche. Sobald der Pull Request erstellt wurde, besteht der angezeigte Unterschied zwischen der Revision des Pull-Requests, den du ausgewählt hast, und dem Commit, der bei der Erstellung des Pull-Requests der Tip des Ziel-Branche war. Weitere Informationen zu Unterschieden und Merge-Bases in Git finden Sie [git-merge-base](#) in der Git-Dokumentation.

Sie können beim Erstellen von Pull Requests die Funktion Beschreibung für mich schreiben verwenden, damit Amazon Q automatisch eine Beschreibung der in einer Pull-Anfrage enthaltenen Änderungen erstellt. Wenn Sie diese Option wählen, analysiert Amazon Q die Unterschiede zwischen dem Quell-Branch, der die Codeänderungen enthält, und dem Ziel-Branch, in dem Sie diese Änderungen zusammenführen möchten. Anschließend erstellt es eine Zusammenfassung der Änderungen sowie die beste Interpretation der Absicht und Wirkung dieser Änderungen.

### Note

Bereitgestellt von Amazon Bedrock: AWS implementiert [automatisierte Missbrauchserkennung](#). Da die Funktionen Beschreibung für mich schreiben und Inhaltzusammenfassung erstellen auf Amazon Bedrock basieren, können Benutzer die in Amazon Bedrock implementierten Kontrollen in vollem Umfang nutzen, um Sicherheit und den verantwortungsvollen Umgang mit künstlicher Intelligenz (KI) durchzusetzen.

Um eine Pull-Anfrage zu erstellen

1. Navigiere zu deinem Projekt.
2. Führen Sie eine der folgenden Aktionen aus:
  - Wählen Sie im Navigationsbereich Code, dann Pull Requests und dann Create Pull Request aus.
  - Wählen Sie auf der Repository-Startseite die Option Mehr und dann Pull-Request erstellen aus.


- Wählen Sie auf der Projektseite die Option Pull-Request erstellen aus.
3. Stellen Sie im Quell-Repository sicher, dass das angegebene Quell-Repository das ist, das den festgeschriebenen Code enthält. Diese Option wird nur angezeigt, wenn Sie den Pull-Request nicht von der Hauptseite des Repositories aus erstellt haben.
  4. Wählen Sie im Ziel-Branch den Branch aus, in den der Code nach der Überprüfung zusammengeführt werden soll.
  5. Wählen Sie im Quellzweig den Zweig aus, der den festgeschriebenen Code enthält.
  6. Geben Sie im Feld Titel der Pull-Anfrage einen Titel ein, der anderen Benutzern hilft zu verstehen, was überprüft werden muss und warum.
  7. (Optional) Geben Sie unter Beschreibung des Pull-Requests Informationen wie einen Link zu Problemen oder eine Beschreibung Ihrer Änderungen ein.

 Tip

Du kannst Beschreibung für mich schreiben wählen, damit CodeCatalyst automatisch eine Beschreibung der in der Pull-Anfrage enthaltenen Änderungen generiert wird. Sie können Änderungen an der automatisch generierten Beschreibung vornehmen, nachdem Sie sie dem Pull Request hinzugefügt haben.

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).

8. (Optional) Wählen Sie unter Probleme die Option Probleme verknüpfen aus und wählen Sie dann entweder ein Problem aus der Liste aus oder geben Sie dessen ID ein. Um die Verknüpfung eines Problems aufzuheben, wählen Sie das Symbol „Verknüpfung aufheben“.
9. (Optional) Wählen Sie unter Erforderliche Prüfer die Option Erforderliche Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Erforderliche Prüfer müssen die Änderungen genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.


 Note

Sie können einen Prüfer nicht gleichzeitig als erforderlichen Prüfer und als optionalen Prüfer hinzufügen. Sie können sich nicht selbst als Prüfer hinzufügen.

10. (Optional) Wählen Sie unter Optionale Prüfer die Option Optionale Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Optionale Prüfer

müssen die Änderungen nicht als Anforderung genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.


11. Überprüfe die Unterschiede zwischen den Branches. Der Unterschied, der in einem Pull Request angezeigt wird, besteht in den Änderungen zwischen der Revision im Quell-Branch und der Merge-Base, bei der es sich um den Head-Commit des Ziel-Branches zum Zeitpunkt der Erstellung des Pull-Requests handelt. Wenn keine Änderungen angezeigt werden, sind die Branches möglicherweise identisch, oder Sie haben denselben Branch sowohl für die Quelle als auch für das Ziel ausgewählt.
12. Wenn Sie überzeugt sind, dass der Pull Request den Code und die Änderungen enthält, die Sie überprüfen möchten, wählen Sie Create aus.

 Note

Nachdem Sie den Pull Request erstellt haben, können Sie Kommentare hinzufügen. Kommentare können zum Pull-Request oder zu einzelnen Zeilen in Dateien sowie zum gesamten Pull-Request hinzugefügt werden. Du kannst Links zu Ressourcen wie Dateien hinzufügen, indem du das @-Zeichen gefolgt vom Namen der Datei verwendest.


Um einen Pull-Request von einem Branch aus zu erstellen

1. Navigiere zu dem Projekt, in dem du einen Pull Request erstellen möchtest.
2. Wählen Sie im Navigationsbereich Quell-Repositories und wählen Sie dann das Repository aus, das den Branch enthält, in dem Sie Codeänderungen überprüfen möchten.
3. Wählen Sie den Dropdown-Pfeil neben dem Standard-Branch-Namen und wählen Sie dann den gewünschten Branch aus der Liste aus. Um alle Branches für ein Repository anzuzeigen, wählen Sie Alle anzeigen.
4. Wähle Mehr und dann Pull-Request erstellen aus.
5. Das Repository und der Quell-Branch sind für Sie vorausgewählt. Wählen Sie im Zielzweig den Zweig aus, in dem Sie den Code zusammenführen möchten, sobald er überprüft wurde. Geben Sie im Feld Titel der Pull-Anfrage einen Titel ein, der anderen Projektbenutzern hilft zu verstehen, was überprüft werden muss und warum. Geben Sie optional in der Beschreibung des Pull-Requests weitere Informationen ein, z. B. indem Sie einen Link zu verwandten Problemen in CodeCatalyst einfügen oder eine Beschreibung der von Ihnen vorgenommenen Änderungen hinzufügen.

 Note

Workflows, die so konfiguriert sind, dass sie für Ereignisse zum Erstellen von Pull-Requests ausgeführt werden, werden nach der Erstellung der Pull-Anfrage ausgeführt, wenn der Ziel-Branch für den Pull Request mit einem der im Workflow angegebenen Branches übereinstimmt.

6. Überprüfen Sie die Unterschiede zwischen den Branches. Wenn keine Änderungen angezeigt werden, sind die Zweige möglicherweise identisch, oder Sie haben denselben Zweig sowohl für die Quelle als auch für das Ziel ausgewählt.
7. (Optional) Wählen Sie unter Probleme die Option Probleme verknüpfen aus und wählen Sie dann entweder ein Problem aus der Liste aus, oder geben Sie dessen ID ein. Um die Verknüpfung eines Problems aufzuheben, wählen Sie das Symbol „Verknüpfung aufheben“.
8. (Optional) Wählen Sie unter Erforderliche Prüfer die Option Erforderliche Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Erforderliche Prüfer müssen die Änderungen genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.

 Note

Sie können einen Prüfer nicht sowohl als erforderlich als auch als optional hinzufügen. Sie können sich nicht selbst als Prüfer hinzufügen.

9. (Optional) Wählen Sie unter Optionale Prüfer die Option Optionale Prüfer hinzufügen aus. Wählen Sie aus der Liste der Projektmitglieder aus, um sie hinzuzufügen. Optionale Prüfer müssen die Änderungen nicht genehmigen, bevor der Pull-Request mit dem Ziel-Branch zusammengeführt werden kann.
10. Wenn Sie überzeugt sind, dass der Pull Request die Änderungen, die Sie überprüfen möchten, und die erforderlichen Reviewer enthält, wählen Sie Create aus.

Wenn Sie Workflows so konfiguriert haben, dass sie ausgeführt werden, bei denen der Branch mit dem Ziel-Branch im Pull Request übereinstimmt, werden Ihnen nach der Erstellung des Pull Requests Informationen zu diesen Workflow-Ausführungen in der Übersicht im Bereich Pull-Request-Details angezeigt. Weitere Informationen finden Sie unter [Hinzufügen eines Push-, Pull- oder Zeitplanauslösers](#).

## Pull-Requests anzeigen

Die generativen KI-Funktionen in Amazon CodeCatalyst befinden sich in der Vorschauversion und können sich ändern. Sie sind nur in der Region USA West (Oregon) verfügbar. Der Zugang zu generativen KI-Funktionen ist je nach Stufe unterschiedlich. Weitere Informationen finden Sie unter [-Preisgestaltung](#).

Sie können Pull-Requests für ein Projekt in der CodeCatalyst Amazon-Konsole anzeigen. Auf der Seite mit der Projektzusammenfassung werden alle offenen Pull Requests für ein Projekt angezeigt. Um alle Pull Requests unabhängig von ihrem Status anzuzeigen, navigieren Sie zur Seite mit den Pull Requests für Ihr Projekt. Wenn du dir einen Pull Request ansiehst, kannst du wählen, ob du eine Zusammenfassung aller Kommentare zu den für dich erstellten Änderungen an dem Pull Request erhalten möchtest.

### Note

Bereitgestellt von Amazon Bedrock: AWS implementiert [automatisierte Missbrauchserkennung](#). Da die Funktionen Beschreibung für mich schreiben und Inhaltszusammenfassung erstellen auf Amazon Bedrock basieren, können Benutzer die in Amazon Bedrock implementierten Kontrollen in vollem Umfang nutzen, um Sicherheit und den verantwortungsvollen Umgang mit künstlicher Intelligenz (KI) durchzusetzen.

Um offene Pull-Requests anzusehen


1. Navigiere zu dem Projekt, in dem du Pull Requests ansehen möchtest.
2. Auf der Projektseite werden offene Pull Requests angezeigt, einschließlich Informationen darüber, wer den Pull Request erstellt hat, welches Repository die Branches für den Pull Request enthält und an welchem Datum der Pull Request erstellt wurde. Sie können die Ansicht der geöffneten Pull-Requests nach dem Quell-Repository filtern.
3. Um alle Pull-Requests anzuzeigen, wähle Alle anzeigen. Du kannst die Selektoren verwenden, um zwischen den Optionen zu wählen. Um beispielsweise alle Pull-Requests anzuzeigen, wähle Beliebiger Status und Beliebiger Autor aus.

Sie können auch im Navigationsbereich Code und dann Pull Requests auswählen und dann mithilfe der Selektoren Ihre Ansicht verfeinern.

4. Auf der Seite Pull Requests kannst du Pull Requests nach ID, Titel, Status und mehr sortieren. Um anzupassen, welche Informationen und wie viele Informationen auf der Pull-Requests-Seite angezeigt werden, wähle das Zahnradsymbol.
5. Um einen bestimmten Pull Request anzuzeigen, wähle ihn aus der Liste aus.
6. Um den Status der mit diesem Pull Request verknüpften Workflow-Ausführungen, falls vorhanden, anzuzeigen, wähle Übersicht und überprüfe die Informationen im Bereich Pull-Request-Details des Pull Requests unter Workflow-Läufe.

Ein Workflow-Lauf findet statt, wenn der Workflow mit Ereignissen zum Erstellen oder Überarbeiten von Pull-Requests konfiguriert ist und wenn die Anforderungen für den Ziel-Branch im Workflow mit dem im Pull Request angegebenen Ziel-Branch übereinstimmen. Weitere Informationen finden Sie unter [Hinzufügen eines Push-, Pull- oder Zeitplanauslösers](#).

7. Um gegebenenfalls verknüpfte Probleme anzuzeigen, wählen Sie Übersicht und überprüfen Sie die Informationen in den Pull-Request-Details unter Probleme. Wenn du ein verknüpftes Problem anzeigen möchtest, wähle dessen ID aus der Liste aus.
8. (Optional) Um eine Zusammenfassung der Kommentare zu erstellen, die zu den Codeänderungen in Überarbeitungen des Pull-Requests hinterlassen wurden, wähle Inhaltszusammenfassung erstellen aus. Die Zusammenfassung wird keine Kommentare enthalten, die zum gesamten Pull-Request hinterlassen wurden.

 Note

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Space aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).


9. Um die Codeänderungen im Pull Request anzuzeigen, wähle Changes aus. Unter Geänderte Dateien kannst du schnell einsehen, wie viele Dateien Änderungen im Pull Request enthalten haben und zu welchen Dateien im Pull Request Kommentare hinzugefügt wurden. Die Anzahl der Kommentare, die neben einem Ordner angezeigt werden, gibt die Anzahl der Kommentare zu Dateien in diesem Ordner an. Erweitern Sie den Ordner, um die Anzahl der Kommentare für jede Datei im Ordner anzuzeigen. Sie können sich auch alle Kommentare ansehen, die zu bestimmten Codezeilen hinterlassen wurden.



 Note

Nicht alle Änderungen in einer Pull-Anfrage können in der Konsole angezeigt werden. Sie können beispielsweise keine Git-Submodule in der Konsole anzeigen, sodass Sie auch keine Unterschiede in einem Submodul in einer Pull-Anfrage anzeigen können. Einige Unterschiede sind möglicherweise zu groß, um angezeigt zu werden. Weitere Informationen finden Sie unter [Kontingente für Quell-Repositorys in CodeCatalyst](#) und [Eine Datei ansehen](#).

10. Um Qualitätsberichte für diesen Pull Request anzuzeigen, wähle Berichte aus.

 Note


Ein Workflow muss so konfiguriert sein, dass er Berichte generiert, damit sie in Ihren Pull Requests angezeigt werden. Weitere Informationen finden Sie unter [Testen mit Workflows in CodeCatalyst](#).

## Anforderungen für das Zusammenführen einer Pull-Anfrage mit Genehmigungsregeln verwalten

Wenn du einen Pull Request erstellst, kannst du wählen, ob du diesem einzelnen Pull Request erforderliche oder optionale Reviewer hinzufügen möchtest. Du kannst jedoch auch Anforderungen erstellen, die alle Pull Requests erfüllen müssen, wenn sie mit einem bestimmten Ziel-Branch zusammengeführt werden. Diese Anforderungen werden als Genehmigungsregeln bezeichnet. Genehmigungsregeln werden für Branches in einem Repository konfiguriert. Wenn Sie einen Pull Request erstellen, für dessen Ziel-Branch eine Genehmigungsregel konfiguriert ist, müssen die Anforderungen für diese Regel zusätzlich zu den Genehmigungen aller erforderlichen Prüfer erfüllt sein, bevor Sie den Pull Request mit diesem Branch zusammenführen können. Durch das Erstellen von Genehmigungsregeln können Sie die Qualitätsstandards für Zusammenführungen mit Branches wie Ihrem Standard-Branch aufrechterhalten.

Genehmigungsregeln, die auf den Standard-Branch Ihres Quell-Repositorys angewendet werden, verhalten sich etwas anders als Genehmigungsregeln, die auf andere Branches angewendet werden. Jede Regel, die auf den Standard-Branch angewendet wird, wird automatisch auf jeden Branch angewendet, den Sie als Standard-Branch angeben. Der Zweig, der zuvor als Standardzweig festgelegt wurde, behält weiterhin die Regeln bei, die auf ihn angewendet wurden.

Wenn Sie Genehmigungsregeln erstellen, sollten Sie berücksichtigen, wie diese Regel von Ihren Projektbenutzern sowohl in der Gegenwart als auch in future eingehalten wird. Wenn Sie beispielsweise sechs Benutzer in Ihrem Projekt haben und Sie eine Genehmigungsregel erstellen, die fünf Genehmigungen erfordert, bevor sie mit dem Ziel-Branch zusammengeführt werden kann, haben Sie effektiv eine Regel erstellt, nach der alle außer der Person, die die Pull Requests erstellt hat, diese Pull Request genehmigen müssen, bevor sie zusammengeführt werden kann.

 Note

Sie benötigen die Rolle des Projektadministrators, um Genehmigungsregeln in CodeCatalyst Projekten erstellen und verwalten zu können. Sie können keine Genehmigungsregeln für verknüpfte Repositorys erstellen.

Sie können Genehmigungsregeln nicht löschen, aber Sie können sie so aktualisieren, dass keine Genehmigungen mehr erforderlich sind, wodurch die Regel effektiv entfernt wird.

Um Genehmigungsregeln für Ziel-Branche für Pull Requests anzuzeigen und zu bearbeiten

1. Navigieren Sie zu dem Projekt, in dem sich Ihr Repository befindet.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.

Wählen Sie das Repository aus, in dem Sie die Genehmigungsregeln anzeigen möchten.

3. Wählen Sie auf der Übersichtsseite des Repositorys die Option Branches aus.
4. Wählen Sie in der Spalte Genehmigungsregeln die Option Anzeigen aus, um den Status aller Regeln für jeden Zweig des Repositorys anzuzeigen.

Im Feld Mindestanzahl von Genehmigungen entspricht die Anzahl der Genehmigungen, die erforderlich sind, bevor ein Pull Request mit diesem Branch zusammengeführt werden kann.

5. Um eine Genehmigungsregel zu erstellen oder zu ändern, wählen Sie Einstellungen verwalten aus. Wählen Sie auf der Einstellungsseite für das Quell-Repository unter Genehmigungsregeln die Option Bearbeiten aus.

**Note**

Sie benötigen die Rolle eines Projektadministrators, um Genehmigungsregeln bearbeiten zu können.

- Wählen Sie unter Branch aus der Dropdownliste den Namen der Filiale aus, für die Sie eine Genehmigungsregel konfigurieren möchten. Geben Sie im Feld Mindestanzahl von Genehmigungen eine Zahl ein, und wählen Sie dann Speichern aus.

## Einen Pull-Request überprüfen

Die generativen KI-Funktionen in Amazon CodeCatalyst befinden sich in der Vorschauversion und können sich ändern. Sie sind nur in der Region USA West (Oregon) verfügbar. Der Zugang zu generativen KI-Funktionen ist je nach Stufe unterschiedlich. Weitere Informationen finden Sie unter [-Preisgestaltung](#).

Sie können die CodeCatalyst Amazon-Konsole verwenden, um die in einer Pull-Anfrage enthaltenen Änderungen gemeinsam zu überprüfen und zu kommentieren. Sie können Kommentare zu einzelnen Codezeilen hinzufügen, die sich auf den Unterschied zwischen den Quell- und Zielbranches oder auf den Unterschied zwischen den Versionen des Pull-Requests beziehen. Sie können sich dafür entscheiden, eine Zusammenfassung der Kommentare zu den Codeänderungen im Pull Request zu erstellen, um das Feedback anderer Benutzer schnell zu verstehen. Du kannst dich auch dafür entscheiden, eine Entwicklungsumgebung zu erstellen, um an Code zu arbeiten.

**Note**

Bereitgestellt von Amazon Bedrock: AWS implementiert [automatisierte Missbrauchserkennung](#). Da die Funktionen Beschreibung für mich schreiben und Inhaltszusammenfassung erstellen auf Amazon Bedrock basieren, können Benutzer die in Amazon Bedrock implementierten Kontrollen in vollem Umfang nutzen, um Sicherheit und den verantwortungsvollen Umgang mit künstlicher Intelligenz (KI) durchzusetzen.

 Tip

Sie können als Teil Ihres Profils konfigurieren, über welche Pull-Request-Ereignisse Sie E-Mails erhalten. Weitere Informationen finden Sie unter [Benachrichtigungen in Amazon CodeCatalyst](#).

Pull-Requests zeigen, was der Unterschied zwischen der Revision des Pull-Requests und dem Commit sein wird, das bei der Erstellung des Pull-Requests der Tipp des Ziel-Branche war. Dies wird als Merge-Base bezeichnet. Weitere Informationen zu Unterschieden und Merge-Bases in Git finden Sie [git-merge-base](#) in der Git-Dokumentation.

 Tip

Wenn du in der Konsole arbeitest, insbesondere wenn du eine Zeit lang eine Pull-Anfrage geöffnet hast, solltest du erwägen, deinen Browser zu aktualisieren, um sicherzustellen, dass du die neueste Version für eine Pull-Anfrage zur Verfügung hast, bevor du mit der Überprüfung beginnst.

Um einen Pull-Request in der CodeCatalyst Konsole zu überprüfen

1. Navigiere zu deinem Projekt.
2. Navigiere zu den Pull Requests, indem du einen der folgenden Schritte ausführst:
  - Wenn der Pull Request auf der Projektseite aufgeführt ist, wählen Sie ihn aus der Liste aus.
  - Wenn der Pull Request nicht auf der Projektseite aufgeführt ist, wähle Alle anzeigen aus. Verwende die Filter und die Sortierung, um den Pull Request zu finden, und wähle ihn dann aus der Liste aus.
  - Wählen Sie im Navigationsbereich Code und dann Pull Requests aus.
3. Wählen Sie die Pull-Anfrage, die Sie überprüfen möchten, aus der Liste aus. Du kannst die Liste der Pull-Requests filtern, indem du einen Teil des Namens in die Filterleiste eingibst.
4. In der Übersicht kannst du den Namen und den Titel des Pull Requests überprüfen. Du kannst Kommentare, die zum Pull Request selbst hinterlassen wurden, erstellen und ansehen. Du kannst dir auch die Details des Pull-Requests ansehen, einschließlich Informationen über Workflow-Ausführungen, verknüpfte Probleme, Prüfer, den Autor des Pull-Requests und mögliche Merge-Strategien.

**Note**

Kommentare, die zu bestimmten Codezeilen hinterlassen wurden, werden unter Änderungen angezeigt.

5. (Optional) Um einen Kommentar hinzuzufügen, der für die gesamte Pull-Anfrage gilt, erweitern Sie Kommentare zur Pull-Anfrage und wählen Sie dann Kommentar erstellen aus.
6. (Optional) Um eine Zusammenfassung aller Kommentare zu Änderungen in Revisionen dieses Pull-Requests anzuzeigen, wähle Kommentarzusammenfassung erstellen aus.

**Note**

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).


7. Unter Änderungen kannst du die Unterschiede zwischen dem Ziel-Branch und der neuesten Version des Pull-Requests sehen. Wenn es mehr als eine Revision gibt, kannst du ändern, welche Revisionen im Unterschied zwischen ihnen verglichen werden. Weitere Informationen zu Revisionen finden Sie unter [Überarbeitungen](#).

**Tip**

Unter Geänderte Dateien kannst du schnell einsehen, wie viele Dateien Änderungen im Pull Request enthalten haben und zu welchen Dateien im Pull Request Kommentare hinzugefügt wurden. Die Anzahl der Kommentare, die neben einem Ordner angezeigt werden, gibt die Anzahl der Kommentare zu Dateien in diesem Ordner an. Erweitern Sie den Ordner, um die Anzahl der Kommentare für jede Datei im Ordner anzuzeigen.


8. Um die Art und Weise zu ändern, wie Unterschiede angezeigt werden, wählen Sie zwischen Vereinheitlicht und Geteilt.
9. Um einer Zeile im Pull Request einen Kommentar hinzuzufügen, gehe zu der Zeile, die du kommentieren möchtest. Wählen Sie das Kommentarsymbol, das für diese Zeile angezeigt wird, geben Sie einen Kommentar ein und wählen Sie dann Speichern aus.
10. Um Änderungen zwischen den Versionen eines Pull-Requests oder zwischen dessen Quell- und Ziel-Branches anzuzeigen, wähle eine der verfügbaren Optionen unter Vergleichen aus. Kommentare zu Zeilen in Revisionen bleiben in diesen Revisionen erhalten.

11. Wenn Sie Ihren Workflow so konfiguriert haben, dass ein Bericht zur Codeabdeckung über Pull-Request-Trigger generiert wird, können Sie die Ergebnisse der Leitungs- und Zweigabdeckung im entsprechenden Pull Request einsehen. Um die Ergebnisse der Codeabdeckung auszublenden, wähle „Codeabdeckung ausblenden“. Weitere Informationen finden Sie unter [Berichte zur Codeabdeckung](#).
12. Wenn du Codeänderungen an der Pull-Anfrage vornehmen möchtest, kannst du aus der Pull-Anfrage eine Entwicklungsumgebung erstellen. Wählen Sie Create Dev Environment. Fügen Sie optional einen Namen für die Entwicklungsumgebung hinzu oder bearbeiten Sie deren Konfiguration und wählen Sie dann Create.
13. Unter Berichte kannst du die Qualitätsberichte in diesem Pull-Request einsehen. Wenn es mehr als eine Revision gibt, kannst du ändern, welche Revisionen verglichen werden, was den Unterschied zwischen ihnen angeht. Sie können die Berichte nach Name, Status, Arbeitsablauf, Aktion und Typ filtern.

 Note

Ein Workflow muss so konfiguriert sein, dass er Berichte generiert, damit sie in Ihren Pull-Requests angezeigt werden. Weitere Informationen finden Sie unter [Berichte konfigurieren](#).

14. Um einen bestimmten Bericht anzuzeigen, wählen Sie ihn aus der Liste aus. Weitere Informationen finden Sie unter [Testen mit Workflows in CodeCatalyst](#).
15. Wenn Sie als Prüfer für diesen Pull Request aufgeführt sind und die Änderungen genehmigen möchten, stellen Sie sicher, dass Sie sich die neueste Version ansehen, und wählen Sie dann Genehmigen aus.

 Note

Alle erforderlichen Reviewer müssen einen Pull Request genehmigen, bevor er zusammengeführt werden kann.

## Aktualisierung einer Pull-Anfrage

Du kannst es anderen Projektmitgliedern erleichtern, Code zu überprüfen, indem du den Pull Request aktualisierst. Du kannst einen Pull Request aktualisieren, um seine Reviewer, seine Links zu Issues, den Titel des Pull Requests oder seine Beschreibung zu ändern. Beispielsweise möchtest du

vielleicht die erforderlichen Reviewer für einen Pull Request ändern, um jemanden zu entfernen, der im Urlaub ist, und eine weitere Person hinzuzufügen. Du kannst einen Pull Request auch mit weiteren Codeänderungen aktualisieren, indem du Commits in den Quell-Branch eines geöffneten Pull Requests überträgst. Jeder Push an den Quell-Branch einer Pull-Anfrage im CodeCatalyst Quell-Repository erzeugt eine Revision. Projektmitglieder können die Unterschiede zwischen den Revisionen in einem Pull Request einsehen.

Um die Reviewer für einen Pull Request zu aktualisieren

1. Navigiere zu dem Projekt, in dem du die Reviewer eines Pull Requests aktualisieren möchtest.
2. Wählen Sie auf der Projektseite unter Pull Requests öffnen den Pull Request aus, für den Sie die Reviewer aktualisieren möchten. Sie können auch im Navigationsbereich Code, dann Pull Requests und dann den Pull Request auswählen, den Sie aktualisieren möchten.
3. (Optional) Wählen Sie unter Übersicht im Bereich Pull-Request-Details das Pluszeichen aus, um erforderliche oder optionale Reviewer hinzuzufügen. Wählen Sie das X neben einem Prüfer aus, um ihn als optionalen oder erforderlichen Prüfer zu entfernen.
4. (Optional) Wählen Sie unter Übersicht im Bereich Details zur Pull-Anfrage die Option Probleme verknüpfen aus, um ein Problem mit der Pull-Anfrage zu verknüpfen, und wählen Sie dann entweder ein Problem aus der Liste aus oder geben Sie dessen ID ein. Um die Verknüpfung eines Issues aufzuheben, wähle das Symbol „Verknüpfung aufheben“ neben dem Issue, dessen Verknüpfung du aufheben möchtest.

Um Dateien und Code im Quell-Branch einer Pull-Anfrage zu aktualisieren

1. Um mehrere Dateien zu aktualisieren, [erstellen Sie entweder eine Entwicklungsumgebung](#) oder klonen Sie das Repository und seinen Quellzweig und verwenden Sie einen Git-Client oder eine integrierte Entwicklungsumgebung (IDE), um Änderungen an den Dateien im Quellzweig vorzunehmen. Übernehmen Sie die Änderungen und übertragen Sie sie in den Quell-Branch im CodeCatalyst Quell-Repository, um die Pull-Anfrage automatisch mit den Änderungen zu aktualisieren. Weitere Informationen finden Sie unter [Klonen eines Quell-Repositorys](#) und [Mit Commits in Amazon arbeiten CodeCatalyst](#).
2. Um eine einzelne Datei in einem Quellzweig zu aktualisieren, können Sie einen Git-Client oder eine Git-IDE verwenden, wie Sie es für mehrere Dateien tun würden. Sie können es auch direkt in der CodeCatalyst Konsole bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer Datei](#).

## Um den Titel und die Beschreibung einer Pull-Anfrage zu aktualisieren

1. Navigiere zu dem Projekt, in dem du den Titel oder die Beschreibung eines Pull Requests aktualisieren möchtest.
2. Auf der Projektseite werden offene Pull Requests angezeigt, einschließlich Informationen darüber, wer den Pull Request erstellt hat, welches Repository die Branches für den Pull Request enthält und wann der Pull Request erstellt wurde. Sie können die Ansicht der geöffneten Pull-Requests nach dem Quell-Repository filtern. Wählen Sie den Pull-Request, den Sie ändern möchten, aus der Liste aus.
3. Um alle Pull Requests anzuzeigen, wähle Alle anzeigen. Wählen Sie alternativ im Navigationsbereich Code und dann Pull Requests aus. Verwenden Sie das Filterfeld oder die Sortierfunktionen, um den Pull Request zu finden, den Sie ändern möchten, und wählen Sie ihn dann aus.
4. Wähle in der Übersicht die Option Bearbeiten aus.
5. Ändern Sie den Titel oder die Beschreibung und wählen Sie dann Speichern.

## Eine Pull-Anfrage zusammenführen

Nachdem Ihr Code überprüft und von allen erforderlichen Reviewern genehmigt wurde, können Sie einen Pull-Request in der CodeCatalyst Konsole zusammenführen, indem Sie eine unterstützte Merge-Strategie verwenden, z. B. Fast-Forward. Nicht alle in der CodeCatalyst Konsole unterstützten Merge-Strategien stehen als Auswahlmöglichkeiten für alle Pull-Requests zur Verfügung. CodeCatalyst wertet die Zusammenführung aus und ermöglicht es Ihnen nur, zwischen Merge-Strategien zu wählen, die in der Konsole verfügbar sind und den Quell-Branch mit dem Ziel-Branch zusammenführen können. Du kannst auch eine Pull-Anfrage mit einer Git-Merge-Strategie deiner Wahl zusammenführen, indem du den `git merge` Befehl auf deinem lokalen Computer oder in einer Entwicklungsumgebung ausführst, um den Quell-Branch mit dem Ziel-Branch zusammenzuführen. Sie können diese Änderungen dann im Ziel-Branch in das Quell-Repository in übertragen CodeCatalyst.

### Note

Durch das Zusammenführen des Branches und das Pushen der Änderungen in Git wird der Pull-Request nicht automatisch geschlossen.



Wenn du die Rolle eines Projektadministrators hast, kannst du dich auch dafür entscheiden, einen Pull-Request zusammenzuführen, der noch nicht alle Anforderungen für Genehmigungen und Genehmigungsregeln erfüllt hat.

## Einen Pull-Request zusammenführen (Konsole)

Du kannst einen Pull Request in der CodeCatalyst Konsole zusammenführen, wenn es keine Merge-Konflikte zwischen Quell- und Ziel-Branche gibt und wenn alle erforderlichen Reviewer den Pull Request genehmigt haben. Wenn es Konflikte gibt oder wenn die Zusammenführung nicht abgeschlossen werden kann, ist die Schaltfläche zum Zusammenführen inaktiv und die Bezeichnung Nicht zusammenführbar wird angezeigt. In diesem Fall müssen Sie die Genehmigung aller erforderlichen Genehmiger einholen, Konflikte gegebenenfalls lokal lösen und diese Änderungen übertragen, bevor Sie die Zusammenführung durchführen können. Durch das Zusammenführen eines Pull-Requests wird automatisch eine E-Mail an den Ersteller des Pull-Requests sowie an alle erforderlichen oder optionalen Prüfer gesendet. Probleme im Zusammenhang mit dem Pull-Request werden dadurch nicht automatisch geschlossen oder ihr Status geändert.

### Tip

Du kannst im Rahmen deines Profils konfigurieren, über welche Pull-Request-Ereignisse du E-Mails erhältst. Weitere Informationen finden Sie unter [Benachrichtigungen in Amazon CodeCatalyst](#).

Um einen Pull-Request zusammenzuführen

1. Navigiere zu dem Projekt, in dem du einen Pull Request zusammenführen möchtest.
2. Wähle auf der Projektseite unter Pull Requests öffnen den Pull Request aus, den du zusammenführen möchtest. Wenn du den Pull Request nicht siehst, wähle Alle Pull Requests anzeigen und wähle ihn dann aus der Liste aus. Sie können auch im Navigationsbereich Code, dann Pull Requests und dann den Pull Request auswählen, den Sie zusammenführen möchten. Wählen Sie Merge (Zusammenführen).
3. Wählen Sie aus den verfügbaren Merge-Strategien für den Pull Request aus. Aktivieren oder deaktivieren Sie optional die Option, den Quell-Branch nach dem Zusammenführen des Pull-Requests zu löschen, und wählen Sie dann Merge aus.

**Note**

Wenn die Schaltfläche Zusammenführen inaktiv ist oder dir die Bezeichnung Nicht zusammenführbar angezeigt wird, haben entweder die erforderlichen Prüfer den Pull Request noch nicht genehmigt, oder der Pull Request kann nicht in der Konsole zusammengeführt werden. CodeCatalyst Ein Prüfer, der eine Pull-Anfrage nicht genehmigt hat, wird im Bereich mit den Pull-Request-Details in der Übersicht durch ein Uhrensymbol gekennzeichnet. Wenn alle erforderlichen Reviewer den Pull Request genehmigt haben, die Schaltfläche Zusammenführen aber immer noch inaktiv ist, liegt möglicherweise ein Merge-Konflikt vor. Wähle das unterstrichene Label Nicht zusammenführbar aus, um weitere Informationen darüber zu erhalten, warum der Pull Request nicht zusammengeführt werden kann. Sie können Merge-Konflikte für den Ziel-Branche in einer Entwicklungsumgebung oder in der CodeCatalyst Konsole lösen und dann den Pull-Request zusammenführen, oder Sie können Konflikte lösen und lokal zusammenführen und dann den Commit, der den Merge enthält, in den Quell-Branche übertragen. CodeCatalyst Weitere Informationen finden Sie unter [Eine Pull-Anfrage zusammenführen \(Git\)](#) und in Ihrer Git-Dokumentation.

## Überschreibe die Anforderungen für Zusammenführungen

Wenn Sie die Rolle eines Projektadministrators haben, können Sie einen Pull-Request zusammenführen, der noch nicht alle Anforderungen für die erforderlichen Genehmigungen und Genehmigungsregeln erfüllt hat. Dies wird als Außerkraftsetzen der Anforderungen für einen Pull-Request bezeichnet. Sie können sich dafür entscheiden, wenn ein erforderlicher Prüfer nicht verfügbar ist oder wenn es dringend erforderlich ist, einen bestimmten Pull-Request mit einem Branch zusammenzuführen, dessen Genehmigungsregeln nicht schnell erfüllt werden können.

Um einen Pull Request zusammenzuführen

1. Wähle in der Pull-Anfrage, in der du Anforderungen überschreiben und zusammenführen möchtest, den Dropdown-Pfeil neben der Schaltfläche Zusammenführen aus. Wähle Genehmigungsanforderungen außer Kraft setzen.
2. Geben Sie unter Grund außer Kraft setzen Einzelheiten dazu an, warum Sie diesen Pull-Request zusammenführen, ohne dass er die Genehmigungsregeln und die erforderlichen Prüferanforderungen erfüllt. Dies ist zwar optional, wird aber dringend empfohlen.

3. Wählen Sie optional eine Merge-Strategie oder akzeptieren Sie die Standardeinstellung. Sie können sich auch dafür entscheiden, die automatisch generierte Commit-Nachricht mit weiteren Details zu aktualisieren.
4. Aktivieren oder deaktivieren Sie die Option, mit der der Quell-Branch beim Zusammenführen gelöscht werden soll. Wir empfehlen, den Quell-Branch beizubehalten, wenn Sie die Anforderungen für das Zusammenführen eines Pull-Requests außer Kraft setzen, bis Sie die Möglichkeit hatten, die Entscheidung mit anderen Teammitgliedern zu überprüfen.
5. Wählen Sie Merge (Zusammenführen).

## Eine Pull-Anfrage zusammenführen (Git)

Git unterstützt viele Optionen zum Zusammenführen und Verwalten von Branches. Die folgenden Befehle sind einige der Optionen, die du verwenden kannst. Weitere Informationen finden Sie in der verfügbaren Dokumentation auf der [Git-Website](#). Sobald du deine Änderungen zusammengeführt und übertragen hast, schließe den Pull-Request manuell. Weitere Informationen finden Sie unter [Einen Pull-Request schließen](#).

### Allgemeine Git-Befehle zum Zusammenführen von Branches

Führt Änderungen vom Quell-Branch im lokalen Repo mit dem Ziel-Branch im lokalen Repo zusammen.

```
git checkout destination-branch-name
```

```
git merge source-branch-name
```

Führt den Quellzweig mit dem Zielzweig zusammen und gibt dabei eine Schnellvorwärtszusammenführung an. Dadurch werden die Zweige zusammengeführt und der Zeiger des Zielzweigs an die Spitze des Quellzweigs bewegt.

```
git checkout destination-branch-name
```

```
git merge --ff-only source-branch-name
```

Führt den Quellzweig mit dem Zielzweig zusammen und gibt eine Squash-Merge an. Dadurch werden alle Commits aus dem Quell-Branch zu einem einzigen Merge-Commit im Ziel-Branch zusammengefasst.

```
git checkout destination-branch-name
```

```
git merge --squash source-branch-name
```

Führt den Quell-Branch mit dem Ziel-Branch zusammen und gibt dabei eine Drei-Wege-Merge an. Dadurch wird ein Merge-Commit erstellt und die einzelnen Commits aus dem Quell-Branch zum Ziel-Branch hinzugefügt.

```
git checkout destination-branch-name
```

```
git merge --no-ff source-branch-name
```

Löscht den Quellzweig im lokalen Repository. Dies ist nützlich, um Ihr lokales Repository zu bereinigen, nachdem Sie es mit dem Ziel-Branch zusammengeführt und die Änderungen in das Quell-Repository übertragen haben.

```
git branch -d source-branch-name
```

Löscht den Quell-Branch im Remote-Repository (das Quell-Repository in CodeCatalyst) unter Verwendung des im lokalen Repo angegebenen Nicknamens für das Remote-Repository. (Beachten Sie die Verwendung des Doppelpunkts (:).) Sie können dies auch `--delete` als Teil des Befehls angeben.

```
git push remote-name :source-branch-name
```

```
git push remote-name --delete source-branch-name
```

## Einen Pull-Request schließen

Du kannst eine Pull-Anfrage als Geschlossen markieren. Dadurch wird der Pull-Request nicht zusammengeführt, aber es kann dir helfen festzustellen, welche Pull-Requests eine Aktion erfordern und welche Pull-Requests nicht mehr relevant sind. Wir empfehlen, einen Pull-Request zu schließen, nachdem du ihn zusammengeführt hast. Beim Veröffentlichen eines Pull-Requests wird automatisch eine E-Mail an den Ersteller des Pull-Requests sowie an alle erforderlichen oder optionalen Reviewer gesendet. Der Status von Problemen im Zusammenhang mit dem Pull-Request wird dadurch nicht automatisch geändert.

### Note

Sie können einen Pull-Request nicht erneut öffnen, nachdem er geschlossen wurde.

## Um eine Pull-Anfrage zu schließen

1. Navigiere zu dem Projekt, in dem du einen Pull Request schließen möchtest.
2. Auf der Projektseite werden offene Pull Requests angezeigt. Wählen Sie den Pull Request aus, den Sie schließen möchten.
3. Klicken Sie auf Schließen.
4. Überprüfe die Informationen und wähle dann Pull-Request schließen aus.

## Mit Commits in Amazon arbeiten CodeCatalyst

Commits sind Snapshots der Inhalte und der Änderungen, die an den Inhalten des Repositorys vorgenommen wurden. Jedes Mal, wenn ein Benutzer eine Änderung festschreibt und an einen Branch weitergibt, werden diese Informationen gespeichert. Zu den Git-Commit-Informationen gehören der Commit-Autor, die Person, die die Änderung vorgenommen hat, Datum und Uhrzeit sowie die vorgenommenen Änderungen. Ähnliche Informationen sind automatisch enthalten, wenn Sie eine Datei in der CodeCatalyst Amazon-Konsole erstellen oder bearbeiten, aber der Autorenname ist Ihr CodeCatalyst Benutzername. Du kannst Commits auch Git-Tags hinzufügen, um dir zu helfen, bestimmte Commits zu identifizieren.

Bei Amazon CodeCatalyst können Sie:

- Eine Liste der Commits für einen Branch anzeigen.
- Sehen Sie sich einzelne Commits an, einschließlich der Änderungen, die in einem Commit im Vergleich zu seinem übergeordneten Commit oder seinen übergeordneten Commits vorgenommen wurden.

Sie können auch Dateien und Ordner anzeigen. Weitere Informationen finden Sie unter [Arbeiten mit Dateien in Amazon CodeCatalyst](#).

### Themen

- [Commits für einen Branch anzeigen](#)
- [Ändern der Anzeige von Commits \(CodeCatalystKonsole\)](#)

## Commits für einen Branch anzeigen

Sie können den Verlauf der an einem Branch vorgenommenen Änderungen einsehen, indem Sie die Commits des Branches in der CodeCatalyst Konsole überprüfen. Auf diese Weise können Sie besser nachvollziehen, wer wann Änderungen an dem Branch vorgenommen hat. Sie können auch die Änderungen überprüfen, die in einem bestimmten Commit vorgenommen wurden.

Du kannst Commits auch mit deinem Git-Client anzeigen. Weitere Informationen findest du in deiner Git-Dokumentation.

### Um Commits anzusehen (Konsole)

1. Navigieren Sie zu dem Projekt, das das Quell-Repository enthält, in dem Sie Commits anzeigen möchten.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.

Wählen Sie das Repository aus, in dem Sie Commits für einen Branch anzeigen möchten.

3. Der Standard-Branch des Repositorys wird angezeigt, einschließlich Informationen über den letzten Commit für den Branch. Wählen Sie Commits aus. Wählen Sie alternativ Mehr und dann Commits anzeigen aus.
4. Um Commits für einen anderen Branch anzuzeigen, wählen Sie den Branch-Selektor und dann den Namen des Branches aus.
5. Um Details zu einem bestimmten Commit anzuzeigen, wählen Sie seinen Titel aus dem Feld Commit-Titel aus. Es werden Details zum Commit angezeigt, einschließlich Informationen über den übergeordneten Commit und die Änderungen, die am Code vorgenommen wurden, indem der übergeordnete Commit mit dem angegebenen Commit verglichen wurde.

#### Tip

Wenn ein Commit mehr als einen übergeordneten Commit hat, können Sie auswählen, für welchen übergeordneten Commit Sie Informationen und Änderungen anzeigen möchten, indem Sie das Dropdown-Symbol neben der übergeordneten Commit-ID auswählen.

## Ändern der Anzeige von Commits (CodeCatalystKonsole)

Sie können ändern, welche Informationen in der Commits-Ansicht angezeigt werden. Sie können wählen, ob Spalten wie Autor und Commit-ID ein- oder ausgeblendet werden sollen.

Um zu ändern, wie Commits angezeigt werden (Konsole)

1. Navigieren Sie zu dem Projekt, das das Quell-Repository enthält, in dem Sie Commits anzeigen möchten.
2. Wählen Sie den Namen des Repositorys aus der Liste der Quell-Repositorys für das Projekt aus. Sie können auch im Navigationsbereich Code und dann Quell-Repositorys auswählen.

Wählen Sie das Repository aus, in dem Sie ändern möchten, wie Commits angezeigt werden.

3. Der Standard-Branch des Repositorys wird angezeigt, einschließlich Informationen über den letzten Commit für den Branch. Wählen Sie Commits aus.
4. Wählen Sie das Zahnradsymbol aus.
5. Wählen Sie in den Einstellungen die Anzahl der anzuzeigenden Commits aus und legen Sie fest, ob Informationen zum Commit-Autor, zum Commit-Datum und zur Commit-ID angezeigt werden sollen.

### Note

Sie können den Commit-Titel nicht in der Informationsanzeige ausblenden.

6. Wenn Sie Ihre Änderungen vorgenommen haben, wählen Sie Speichern, um sie zu speichern, oder Abbrechen, um sie zu verwerfen.

## Kontingente für Quell-Repositorys in CodeCatalyst



In der folgenden Tabelle werden Kontingente und Beschränkungen für Quell-Repositorys in Amazon CodeCatalyst beschrieben. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

Ressource	Informationen
Branch-Namen	Jede Kombination von zulässigen Zeichen mit einer Länge zwischen 1 und 256 Zeichen, die

Ressource	Informationen
	<p>innerhalb eines Repositorys eindeutig sein muss. Branch-Namen dürfen nicht:</p> <ul style="list-style-type: none"><li>• mit einem Schrägstrich (/) oder Punkt (.) anfangen oder enden.</li><li>• nur das einzelne Zeichen @ aufweisen.</li><li>• zwei oder mehr aufeinanderfolgende Punkte (.), Schrägstriche (/) oder die folgende Zeichenkombination enthalten: @{</li><li>• Leerzeichen oder eines der folgenden Zeichen enthalten: ? ^ * [ \ ~ :</li></ul> <p>Branch-Namen sind Referenzen. Viele der Einschränkungen für Branch-Namen basieren auf dem Git-Referenzstandard. Weitere Informationen finden Sie unter <a href="#">Git Internals</a> und <a href="#">git-check-ref-format</a>.</p>
Kommentare zu einer Pull-Anfrage	Maximal 1.000 bei einer Pull-Anfrage.
Nachricht übertragen	Maximal 1024 Zeichen.



Ressource	Informationen
Dateipfade	<p>Beliebige Kombination aus zulässigen Zeichen und mit einer Länge zwischen 1 und 4,096 Zeichen. Dateipfade müssen einen eindeutigen Namen haben, in dem die Datei und der genaue Speicherort der Datei angegeben sind. Dateipfade dürfen nicht mehr als 20 Verzeichnisse umfassen. Außerdem gelten für Dateipfade folgende Regeln:</p> <ul style="list-style-type: none"><li>• Sie dürfen keine leeren Zeichenfolgen enthalten.</li><li>• Sie dürfen kein relativer Dateipfad sein.</li><li>• Sie dürfen keine der folgenden Zeichenkombinationen enthalten:      <code>/./</code>      <code>/. ./</code>      <code>//</code></li><li>• Sie dürfen nicht mit einem Schrägstrich oder umgekehrten Schrägstrich (Backslash) enden.</li></ul> <p>Dateinamen und -pfade müssen vollständig qualifiziert sein. Der Dateiname und der Pfad zu einer Datei auf Ihrem lokalen Computer müssen den Standards für das Betriebssystem entsprechen. Verwenden Sie bei der Angabe des Pfads zu einer Datei in einem Repository die Standards für Amazon Linux.</p>
Dateigröße	Maximal 6 MB für jede einzelne Datei bei Verwendung der CodeCatalyst Konsole.

Ressource	Informationen
In der CodeCatalyst Konsole sichtbare Dateigröße	Maximal 6 MB für jede einzelne Datei bei Verwendung der CodeCatalyst Konsole.
Größe des Git-Blobs	Maximal 2 GB <div data-bbox="829 430 1507 982" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"><p> <b>Note</b></p><p>Die Anzahl oder die Gesamtgröße aller Dateien in einem einzelnen Commit ist nicht begrenzt, sofern die Metadaten den Maximalwert von 6 MB nicht übersteigen und ein einzelner Blob nicht größer ist als 2 GB. Als bewährte Methode sollten Sie jedoch mehrere kleinere Commits anstelle eines großen Commits in Betracht ziehen.</p></div>
Metadaten für Commit	Maximal 6 MB für die kombinierten <a href="#">Metadaten für einen Commit</a> (z. B. die Kombination aus Autoreneinformationen, Datum, übergeordneter Commit-Liste und Commit-Nachrichten). <div data-bbox="829 1243 1507 1698" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"><p> <b>Note</b></p><p>Die Anzahl oder die Gesamtgröße aller Dateien in einem einzelnen Commit ist nicht begrenzt, sofern die Daten den Maximalwert von 20 MB nicht übersteigen und eine einzelne Datei nicht größer als 6 MB und ein einzelner Blob nicht größer als 2 GB ist.</p></div>
Anzahl der CodeCatalyst Probleme, die mit einer Pull-Anfrage verknüpft werden können	50

Ressource	Informationen
Anzahl der Jira-Issues, die mit einer Pull-Anfrage verknüpft werden können	50
Anzahl der offenen Pull-Requests in einem Space	Maximal 1.000 für einen CodeCatalyst Amazon-Bereich.
Gesamtzahl der Pull-Anfragen in einem Space	Maximal 10.000 für einen CodeCatalyst Amazon-Bereich.
Anzahl Referenzen für einzelnen Push	Maximal 4 000 (einschließlich Erstellen, Löschen und Aktualisieren). Für die Gesamtanzahl der Referenzen im Repository ist keine Einschränkung vorhanden.
Anzahl der Repositories in einem Space	Maximal 5.000 für einen CodeCatalyst Amazon-Bereich.
Repository-Beschreibungen	Beliebige Zeichenkombination und mit einer Länge zwischen 0 und 1 000 Zeichen. Repository-Beschreibungen sind optional.
Repository-Namen	Repository-Namen müssen innerhalb eines Projekts eindeutig sein. Sie können eine beliebige Kombination aus Buchstaben, Zahlen, Punkten, Unterstrichen und Bindestrichen mit einer Länge von 1 bis 100 Zeichen enthalten. Bei Namen wird nicht zwischen Groß- und Kleinschreibung unterschieden. Repository-Namen dürfen nicht auf .git enden, dürfen keine Leerzeichen enthalten und dürfen keines der folgenden Zeichen enthalten: ! ? @ # \$ % ^ & * ( ) + = { } [ ]   \ / > < ~ ` ' " ; :

Ressource	Informationen
Größe des Repositoriums	Die Größe des Repositorys wird von den allgemeinen Speicherlimits für Ihren Speicherplatz beeinflusst. Weitere Informationen finden Sie unter <a href="#">-Preise</a> und <a href="#">Behebung von Problemen mit Quell-Repositorys</a> .
Prüfer für einen Pull-Request	Insgesamt maximal 100 Reviewer (optional oder erforderlich) für einen Pull-Request.
Schriftliche Zusammenfassungen für Pull-Requests	Die maximale Anzahl schriftlicher Zusammenfassungen für Pull Requests hängt von der Abrechnungsstufe für deinen Bereich ab. Weitere Informationen finden Sie unter <a href="#">- Preise</a> .

# Entwicklungsumgebungen in CodeCatalyst

Entwicklungsumgebungen sind cloudbasierte Entwicklungsumgebungen. In Amazon verwenden Sie Dev Environments CodeCatalyst, um an dem Code zu arbeiten, der in den Quell-Repositorys Ihres Projekts gespeichert ist. Beim Erstellen von Entwicklungsumgebungen haben Sie mehrere Möglichkeiten:

- Erstellen Sie eine projektspezifische Entwicklungsumgebung CodeCatalyst, um mit einer unterstützten integrierten Entwicklungsumgebung (IDE) an Code zu arbeiten.
- Erstellen Sie eine leere Entwicklungsumgebung, klonen Sie Code aus einem Quell-Repository hinein und bearbeiten Sie diesen Code mit einer unterstützten IDE.
- Erstellen Sie eine Entwicklungsumgebung in einer IDE und klonen Sie ein Quell-Repository in die Entwicklungsumgebung

Eine Dev-Datei ist eine offene Standard-YAML-Datei, die Ihre Entwicklungsumgebungen standardisiert. Mit anderen Worten, diese Datei kodifiziert die erforderlichen Entwicklungstools für Ihre Entwicklungsumgebung. Auf diese Weise können Sie schnell eine Entwicklungsumgebung einrichten, zwischen Projekten wechseln und die Konfiguration der Entwicklungsumgebung für alle Teammitglieder replizieren. Entwicklungsumgebungen minimieren den Zeitaufwand für die Erstellung und Pflege einer lokalen Entwicklungsumgebung, da sie eine Dev-Datei verwenden, die alle Tools konfiguriert, die Sie zum Codieren, Testen und Debuggen für ein bestimmtes Projekt benötigen.

Die in Ihrer Entwicklungsumgebung enthaltenen Projekttools und Anwendungsbibliotheken werden durch die Dev-Datei im Quell-Repository Ihres Projekts definiert. Wenn Sie keine Dev-Datei in Ihrem Quell-Repository haben, CodeCatalyst wird automatisch eine Standard-Dev-Datei angewendet. Diese Standard-Devdatei enthält Tools für die am häufigsten verwendeten Programmiersprachen und Frameworks. Wenn Ihr Projekt mit einem Blueprint erstellt wurde, wird automatisch eine Dev-Datei von erstellt. CodeCatalyst [Weitere Informationen zur Dev-Datei finden Sie unter https://devfile.io](https://devfile.io).

Nachdem Sie eine Entwicklungsumgebung erstellt haben, können nur Sie darauf zugreifen. In Ihrer Entwicklungsumgebung können Sie den Code Ihres Quell-Repositorys in einer unterstützten IDE anzeigen und bearbeiten.

Standardmäßig ist eine Entwicklungsumgebung so konfiguriert, dass sie über einen 2-Core-Prozessor, 4 GB RAM und 16 GB persistenten Speicher verfügt. Wenn Sie über Space-Administratorrechte verfügen, können Sie die Abrechnungsstufe für Ihren Space ändern, um

verschiedene Konfigurationsoptionen für die Entwicklungsumgebung zu verwenden und Rechen- und Speicherlimits zu verwalten. Hier ist ein möglicher Arbeitsablauf für eine Entwicklungsumgebung:

Akua Mansa ist eine neue Entwicklerin bei Example Corp. Akua trat ihrem neuen Team bei, kurz bevor eine neue Version des Teamprodukts veröffentlicht wurde. Das Team von Akua benötigt sie, um umgehend an einer neuen Funktion für die kommende Version des Produkts zu arbeiten. Um einen aufwändigen Einrichtungsprozess zu vermeiden, nimmt Akua eine Einladung an, am Projekt ihres CodeCatalyst Teams teilzunehmen. Anschließend weist sie sich selbst das entsprechende Problem zu und erstellt eine Entwicklungsumgebung aus einem vorhandenen Zweig des Quell-Repositorys ihres Teams. Die Entwicklungsumgebung von Akua öffnet den Code des Quell-Repositorys in der von ihr ausgewählten IDE. Die offene IDE-Instanz ist mit ihrer Entwicklungsumgebung verbunden, die eine Entwicklungsdatei enthält, die automatisch identifiziert und angewendet wurde. Die Dev-Datei spezifiziert alle Tools, die sie für den Einstieg benötigt. Akua schreibt den Code für die neue Produktfunktion, überträgt ihre Codeänderungen, überträgt ihre Änderungen in den bestehenden Zweig und löscht dann ihre Entwicklungsumgebung, wenn ihre Arbeit abgeschlossen ist. Akua hat ohne langwierigen Einrichtungsprozess Code zum Quell-Repository ihres neuen Teams beigetragen.

## Erstellen einer Entwicklungsumgebung

Sie können eine Entwicklungsumgebung auf verschiedene Arten erstellen:

- Erstellen Sie auf den Seiten Übersicht, Entwicklungsumgebungen oder CodeCatalyst Quell-Repositorys eine Entwicklungsumgebung CodeCatalyst mit einem [Quell-Repository oder einem verknüpften](#) Quell-Repository
- Erstellen Sie auf der Seite Entwicklungsumgebungen eine leere Entwicklungsumgebung in CodeCatalyst, die nicht mit einem Quell-Repository verbunden ist
- Erstellen Sie eine Entwicklungsumgebung in der IDE Ihrer Wahl und klonen Sie ein beliebiges Quell-Repository in die Entwicklungsumgebung

Sie können eine Entwicklungsumgebung pro Zweig eines Repositorys erstellen. Ein Projekt kann mehrere Repositorys enthalten. Die von Ihnen erstellten Entwicklungsumgebungen können nur mit Ihrem CodeCatalyst Konto verwaltet werden. Sie können jedoch die Entwicklungsumgebung öffnen und mit jeder der unterstützten IDEs darin arbeiten. Sie müssen die Entwicklungsumgebungen AWS Toolkit installiert haben, um die Entwicklungsumgebungen in Ihrer IDE verwenden zu können. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für](#)

[Entwicklungsumgebungen](#). Standardmäßig werden Entwicklungsumgebungen mit einem 2-Core-Prozessor, 4 GB RAM und 16 GB persistentem Speicher erstellt.

#### Note

Wenn Sie eine Entwicklungsumgebung erstellt haben, die mit einem Quell-Repository verknüpft ist, zeigt die Spalte Ressource immer den Zweig an, den Sie bei der Erstellung dieser Entwicklungsumgebung angegeben haben. Dies gilt auch dann, wenn Sie einen weiteren Branch erstellen, zu einem anderen Branch innerhalb der Entwicklungsumgebung wechseln oder ein zusätzliches Repository klonen. Wenn Sie eine leere Entwicklungsumgebung erstellt haben, ist die Spalte Ressource leer.

## Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen

Sie können Entwicklungsumgebungen mit den folgenden unterstützten integrierten Entwicklungsumgebungen (IDEs) verwenden:

- [AWS Cloud9](#)
- [JetBrains IDEs](#)
  - [IntelliJ IDEA Ultimate](#)
  - [GoLand](#)
  - [PyCharm Professionell](#)
- [Visual Studio-Code](#)


## Erstellen einer Entwicklungsumgebung in CodeCatalyst

Um mit der Arbeit mit Dev Environment in zu beginnen CodeCatalyst, authentifizieren Sie sich und melden Sie sich entweder mit Ihrer [AWS Builder-ID](#) oder [SSO](#) an.

So erstellen Sie eine Entwicklungsumgebung aus einem Branch


1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung erstellen möchten.
3. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:

- Wählen Sie Überblick und navigieren Sie dann zum Abschnitt Meine Entwicklungsumgebungen.
  - Wählen Sie Code und dann Entwicklungsumgebungen aus.
  - Wählen Sie Code, wählen Sie Quell-Repositories und wählen Sie das Repository aus, für das Sie eine Entwicklungsumgebung erstellen möchten.
4. Wählen Sie Entwicklungsumgebung erstellen.
  5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
  6. Wählen Sie Repository klonen.
  7. Führen Sie eine der folgenden Aktionen aus:
    - a. Wählen Sie das zu klonende Repository aus, wählen Sie In vorhandenem Zweig arbeiten und wählen Sie dann einen Zweig aus dem Drop-down-Menü Bestehender Zweig aus.

 Note

Wenn Sie sich für ein Drittanbieter-Repository entscheiden, müssen Sie in einem vorhandenen Branch arbeiten.

- b. Wählen Sie das zu klonende Repository aus, wählen Sie In neuem Zweig arbeiten aus, geben Sie einen Branch-Namen in das Feld Branch-Name ein und wählen Sie aus dem Drop-down-Menü Branch erstellen einen Branch aus, aus dem der neue Branch erstellt werden soll.

 Note


Wenn Sie eine Entwicklungsumgebung von der Seite Quell-Repositorys oder von einem bestimmten Quell-Repository aus erstellen, müssen Sie kein Repository auswählen. Die Entwicklungsumgebung wird aus dem Quell-Repository erstellt, das Sie auf der Seite Quell-Repositorys ausgewählt haben.

8. (Optional) Geben Sie unter Alias — optional einen Alias für die Entwicklungsumgebung ein.
9. (Optional) Wählen Sie die Schaltfläche „Konfiguration bearbeiten“, um die Rechen-, Speicher- oder Timeout-Konfiguration der Entwicklungsumgebung zu bearbeiten.



10. (Optional) Wählen Sie in Amazon Virtual Private Cloud (Amazon VPC) — optional aus dem Drop-down-Menü eine VPC-Verbindung aus, die Sie Ihrer Entwicklungsumgebung zuordnen möchten.

Wenn eine Standard-VPC für Ihren Bereich festgelegt ist, werden Ihre Entwicklungsumgebungen mit dieser VPC verbunden ausgeführt. Sie können dies überschreiben, indem Sie eine andere VPC-Verbindung zuordnen. Beachten Sie außerdem, dass VPC-verbundene Entwicklungsumgebungen dies nicht unterstützen. AWS Toolkit

 Note

Wenn Sie eine Entwicklungsumgebung mit einer VPC-Verbindung erstellen, wird innerhalb der VPC eine neue Netzwerkschnittstelle erstellt. CodeCatalyst interagiert mit dieser Schnittstelle über die zugehörige VPC-Rolle. Stellen Sie außerdem sicher, dass Ihr IPv4-CIDR-Block nicht für den **172.16.0.0/12** IP-Adressbereich konfiguriert ist.


11. Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, wird in der Statusspalte Entwicklungsumgebung der Wert Wird gestartet angezeigt, und in der Statusspalte wird nach der Erstellung der Entwicklungsumgebung die Meldung Wird ausgeführt angezeigt.

Um eine leere Entwicklungsumgebung zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung erstellen möchten.
3. Führen Sie im Navigationsbereich einen der folgenden Schritte aus:
  - Wählen Sie Überblick und navigieren Sie dann zum Abschnitt Meine Entwicklungsumgebungen.
  - Wählen Sie Code und dann Entwicklungsumgebungen aus.
4. Wählen Sie Entwicklungsumgebung erstellen aus.
5. Wählen Sie eine unterstützte IDE aus dem Drop-down-Menü aus. Weitere Informationen finden Sie unter [Unterstützte integrierte Entwicklungsumgebungen für Entwicklungsumgebungen](#).
6. Wählen Sie „Leere Entwicklungsumgebung erstellen“.
7. (Optional) Geben Sie unter Alias — optional einen Alias für die Entwicklungsumgebung ein.
8. (Optional) Wählen Sie die Schaltfläche „Konfiguration bearbeiten“, um die Rechen-, Speicher- oder Timeout-Konfiguration der Entwicklungsumgebung zu bearbeiten.


9. (Optional) Wählen Sie in Amazon Virtual Private Cloud (Amazon VPC) — optional aus dem Drop-down-Menü eine VPC-Verbindung aus, die Sie Ihrer Entwicklungsumgebung zuordnen möchten.

Wenn eine Standard-VPC für Ihren Bereich festgelegt ist, werden Ihre Entwicklungsumgebungen mit dieser VPC verbunden ausgeführt. Sie können dies überschreiben, indem Sie eine andere VPC-Verbindung zuordnen. Beachten Sie außerdem, dass VPC-verbundene Entwicklungsumgebungen dies nicht unterstützen. AWS Toolkit


 Note

Wenn Sie eine Entwicklungsumgebung mit einer VPC-Verbindung erstellen, wird innerhalb der VPC eine neue Netzwerkschnittstelle erstellt. CodeCatalyst interagiert mit dieser Schnittstelle über die zugehörige VPC-Rolle. Stellen Sie außerdem sicher, dass Ihr IPv4-CIDR-Block nicht für den **172.16.0.0/12** IP-Adressbereich konfiguriert ist.

10. Wählen Sie Erstellen. Während Ihre Entwicklungsumgebung erstellt wird, wird in der Statusspalte Entwicklungsumgebung der Wert Wird gestartet angezeigt, und in der Statusspalte wird nach der Erstellung der Entwicklungsumgebung die Meldung Wird ausgeführt angezeigt.

 Note

Das erstmalige Erstellen und Öffnen einer Entwicklungsumgebung kann ein bis zwei Minuten dauern.

 Note

Nachdem die Entwicklungsumgebung in der IDE geöffnet wurde, müssen Sie möglicherweise das Verzeichnis in das Quell-Repository ändern, bevor Sie Änderungen an Ihrem Code übertragen und per Push übertragen.

## Eine Entwicklungsumgebung in einer IDE erstellen

Sie können eine unterstützte integrierte Entwicklungsumgebung (IDE) verwenden, um mit Code zu arbeiten, der in den Quell-Repositorys Ihres Projekts gespeichert ist.

- [Zusammenarbeit mit Amazon CodeCatalyst in AWS Cloud9](#)
- [Amazon CodeCatalyst im VS-Code](#)

Anweisungen zur Authentifizierung finden Sie unter [Authentifizieren und Herstellen einer Verbindung zu AWS Ihrer Entwicklungsumgebung](#) und [Authentifizierung für Amazon CodeCatalyst im AWS Toolkit for Visual Studio Code Benutzerhandbuch](#).

- [Amazon CodeCatalyst in JetBrains](#)

Anweisungen zur Authentifizierung finden Sie unter [Authentifizieren und Verbinden von JetBrains Gateway mit CodeCatalyst](#) im AWS Toolkit for JetBrains Benutzerhandbuch.


## Anhalten einer Entwicklungsumgebung

Das `/projects` Verzeichnis einer Entwicklungsumgebung speichert die Dateien, die aus dem Quell-Repository abgerufen werden, und die Dev-Datei, die zur Konfiguration der Entwicklungsumgebung verwendet wird. Das `/home` Verzeichnis, das bei der Erstellung der Entwicklungsumgebung leer ist, speichert die Dateien, die Sie während der Verwendung Ihrer Entwicklungsumgebung erstellen. Alles in den `/home` Verzeichnissen `/projects` und in den Verzeichnissen einer Entwicklungsumgebung wird persistent gespeichert, sodass Sie die Arbeit in einer Entwicklungsumgebung beenden können, wenn Sie zu einer anderen Entwicklungsumgebung, einem anderen Repository oder einem anderen Projekt wechseln müssen.

### Warning

In einer Entwicklungsumgebung tritt kein Timeout auf, wenn Instanzen, einschließlich Webbrowser, Remote-Shells und IDEs, weiterhin verbunden sind. Stellen Sie also sicher, dass Sie alle verbundenen Instanzen schließen, um zusätzliche Kosten zu vermeiden.


Eine Entwicklungsumgebung wird automatisch beendet, wenn sie für den Zeitraum inaktiv ist, der bei der Erstellung der Entwicklungsumgebung in den Timeout-Feldern ausgewählt wurde. Sie können die Entwicklungsumgebung beenden, bevor sie inaktiv wird. Wenn Sie bei der Erstellung Ihrer Entwicklungsumgebung „Kein Timeout“ ausgewählt haben, wird die Entwicklungsumgebung nicht automatisch beendet. Stattdessen wird sie kontinuierlich ausgeführt.

 Warning

Wenn Sie eine Entwicklungsumgebung beenden, die mit einer gelöschten VPC-Verbindung verknüpft ist, kann sie nicht wieder aufgenommen werden.

So beenden Sie eine Entwicklungsumgebung auf der Seite Entwicklungsumgebungen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung beenden möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Dev Environments aus.
5. Wählen Sie das Optionsfeld für die Entwicklungsumgebung, die Sie beenden möchten.
6. Wählen Sie im Menü „Aktionen“ die Option „Stopp“.

 Note

Die Computernutzung wird nur in Rechnung gestellt, solange die Entwicklungsumgebung läuft. Die Speichernutzung wird jedoch für die gesamte Dauer des Bestehens der Entwicklungsumgebung in Rechnung gestellt. Stoppen Sie Ihre Entwicklungsumgebung, wenn sie nicht verwendet wird, um die Abrechnung der Rechenleistung zu beenden.

## Fortsetzen einer Entwicklungsumgebung

Das `/projects` Verzeichnis einer Entwicklungsumgebung speichert die Dateien, die aus dem Quell-Repository abgerufen werden, und die Dev-Datei, die zur Konfiguration der Entwicklungsumgebung verwendet wird. Das `/home` Verzeichnis, das bei der Erstellung der Entwicklungsumgebung leer ist, speichert die Dateien, die Sie während der Verwendung Ihrer Entwicklungsumgebung erstellen. Alles in den `/home` Verzeichnissen `/projects` und in den Verzeichnissen einer Entwicklungsumgebung wird persistent gespeichert, sodass Sie die Arbeit in einer Entwicklungsumgebung beenden können, wenn Sie zu einer anderen Entwicklungsumgebung, einem anderen Repository oder einem anderen Projekt wechseln müssen, und die Arbeit in Ihrer Entwicklungsumgebung zu einem späteren Zeitpunkt fortsetzen müssen.

Eine Entwicklungsumgebung wird automatisch beendet, wenn sie für den Zeitraum inaktiv ist, der bei der Erstellung der Entwicklungsumgebung in den Timeout-Feldern ausgewählt wurde. Sie müssen den AWS Cloud9 Browser-Tab schließen, damit die Entwicklungsumgebung inaktiv ist.

### Note

Die Entwicklungsumgebung ist weiterhin verfügbar und läuft auch dann, wenn Sie den Zweig löschen, mit dem Sie die Entwicklungsumgebung erstellt haben. Wenn Sie die Arbeit in einer Entwicklungsumgebung fortsetzen möchten, für die Sie den Branch gelöscht haben, erstellen Sie einen neuen Branch und übertragen Sie Ihre Änderungen dorthin.

Um eine Entwicklungsumgebung von der Übersichtsseite aus wieder aufzunehmen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung fortsetzen möchten, und navigieren Sie zum Abschnitt Meine Entwicklungsumgebungen.
3. Wählen Sie Fortsetzen in (IDE).
  - Wählen Sie für JetBrains IDEs JetBrains Gateway-EAP, wenn Sie aufgefordert werden, eine Anwendung zum Öffnen des -Gateway-Links auszuwählen. Wählen Sie Link öffnen, um zu bestätigen, wenn Sie dazu aufgefordert werden.
  - Wählen Sie für die VS Code-IDE die Option VS Code, wenn Sie aufgefordert werden, eine Anwendung zum Öffnen des VS Code-Links auszuwählen. Wählen Sie zur Bestätigung Link öffnen.

Um eine Entwicklungsumgebung vom Quell-Repository aus fortzusetzen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung fortsetzen möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Source Repositories aus.
5. Wählen Sie das Quell-Repository aus, das die Entwicklungsumgebung enthält, die Sie fortsetzen möchten.
6. Wählen Sie den Branch-Namen, um ein Drop-down-Menü mit Ihren Branches anzuzeigen, und wählen Sie dann Ihren Branch aus.

## 7. Wählen Sie Resume Dev Environment.

- Wählen Sie für JetBrains IDEs „Link öffnen“, um zu bestätigen, wenn Sie aufgefordert werden, dieser Site das Öffnen des JetBrains -gateway-Links mit JetBrains Gateway zu erlauben? .
- Wählen Sie für die VS Code-IDE die Option Link öffnen, um zu bestätigen, wenn Sie aufgefordert werden, dieser Site das Öffnen des VS Code-Links mit Visual Studio Code zu erlauben? .

So setzen Sie eine Entwicklungsumgebung auf der Seite Entwicklungsumgebungen fort

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung fortsetzen möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Dev Environments aus.
5. Wählen Sie in der Spalte IDE die Option Resume in (IDE) für die Entwicklungsumgebung aus.
  - Wählen Sie für JetBrains IDEs „Link öffnen“, um zu bestätigen, wenn Sie aufgefordert werden, dieser Site das Öffnen des JetBrains -gateway-Links mit JetBrains Gateway zu erlauben? .
  - Wählen Sie für die VS Code-IDE die Option Link öffnen, um zu bestätigen, wenn Sie aufgefordert werden, dieser Site das Öffnen des VS Code-Links mit Visual Studio Code zu erlauben? .

### Note

Es kann einige Minuten dauern, bis eine Entwicklungsumgebung fortgesetzt wird.

## Eine Entwicklungsumgebung bearbeiten

Während Ihre IDE läuft, können Sie die Entwicklungsumgebung bearbeiten. Wenn Sie Ihr Rechen- oder Inaktivitäts-Timeout bearbeiten, wird Ihre Entwicklungsumgebung neu gestartet, nachdem Sie Ihre Änderungen gespeichert haben.

Um eine Entwicklungsumgebung zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung bearbeiten möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Dev Environments aus.
5. Wählen Sie die Entwicklungsumgebung aus, die Sie bearbeiten möchten.
6. Wählen Sie Bearbeiten aus.
7. Nehmen Sie die gewünschten Änderungen am Timeout für Berechnung oder Inaktivität vor.
8. Klicken Sie auf Speichern.

## Löschen einer Entwicklungsumgebung

Wenn Sie mit der Arbeit an den Inhalten fertig sind, die in Ihrer Entwicklungsumgebung gespeichert sind, können Sie die Entwicklungsumgebung löschen. Erstellen Sie eine neue Entwicklungsumgebung, um an neuen Inhalten zu arbeiten. Wenn Sie Ihre Entwicklungsumgebung löschen, wird der persistente Inhalt dauerhaft gelöscht. Bevor Sie Ihre Entwicklungsumgebung löschen, stellen Sie sicher, dass Sie Ihre Codeänderungen übertragen und in das ursprüngliche Quell-Repository der Entwicklungsumgebung übertragen. Nachdem Sie Ihre Entwicklungsumgebung gelöscht haben, wird die Rechnungs- und Speicherabrechnung für die Entwicklungsumgebung beendet.

Nachdem Sie Ihre Entwicklungsumgebung gelöscht haben, kann es einige Minuten dauern, bis das Speicherkontingent aktualisiert wird. Wenn Sie das Speicherkontingent erreicht haben, können Sie während dieser Zeit keine neue Entwicklungsumgebung erstellen.

### Important

Das Löschen einer Entwicklungsumgebung kann nicht rückgängig gemacht werden. Nachdem Sie eine Entwicklungsumgebung gelöscht haben, können Sie sie nicht mehr wiederherstellen.

Um eine Entwicklungsumgebung zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie eine Entwicklungsumgebung löschen möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Dev Environments aus.

5. Wählen Sie die Entwicklungsumgebung aus, die Sie löschen möchten.
6. Wählen Sie Löschen aus.
7. Geben Sie **eindelete**, um das Löschen der Entwicklungsumgebung zu bestätigen.
8. Wählen Sie Löschen aus.

#### Note

Bevor Sie eine VPC-Verbindung in Ihrem Space löschen, stellen Sie sicher, dass Sie die dieser VPC zugeordnete Entwicklungsumgebung entfernen.

Selbst wenn Sie eine Entwicklungsumgebung löschen, löschen Sie möglicherweise nicht die Netzwerkschnittstelle in der VPC. Stellen Sie sicher, dass Sie Ihre Ressourcen nach Bedarf bereinigen. Wenn beim Löschen einer VPC-verbundenen Entwicklungsumgebung ein Fehler auftritt, müssen Sie Ihre veraltete Verbindung [trennen](#) und sie [löschen](#), nachdem Sie bestätigt haben, dass sie nicht verwendet wird.

## Verbindung zu einer Entwicklungsumgebung über SSH herstellen

Sie können über SSH eine Verbindung zu Ihrer Entwicklungsumgebung herstellen, um Aktionen wie Portweiterleitung, das Hoch- und Herunterladen von Dateien und die Verwendung anderer IDEs ohne Einschränkungen auszuführen.

#### Note

Wenn Sie SSH nach dem Schließen der IDE-Registerkarte oder des IDE-Fensters für eine längere Zeit weiter verwenden möchten, stellen Sie sicher, dass Sie ein hohes Timeout für Ihre Entwicklungsumgebung festlegen, damit sie nicht aufgrund von Inaktivität in der IDE beendet wird.

### Voraussetzungen

- Sie benötigen eines der folgenden Betriebssysteme:
  - Windows 10 oder neuer und OpenSSH aktiviert
  - macOS und Bash Version 3 oder höher
  - Linux mit yum, dpkg oder rpm Paketmanagern und Bash Version 3 oder höher



- Sie benötigen außerdem AWS CLI Version 2.9.4 oder höher.

Um über SSH eine Verbindung zu einer Entwicklungsumgebung herzustellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, für das Sie sich über SSH mit einer Entwicklungsumgebung verbinden möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Dev Environments aus.
5. Wählen Sie eine laufende Entwicklungsumgebung aus, mit der Sie sich über SSH verbinden möchten.
6. Wählen Sie Connect via SSH, wählen Sie Ihr gewünschtes Betriebssystem und gehen Sie wie folgt vor:
  - Falls Sie dies noch nicht getan haben, fügen Sie den ersten Befehl in das angegebene Terminal ein und führen Sie ihn aus. Der Befehl lädt ein Skript herunter und führt die folgenden Änderungen in Ihrer lokalen Umgebung aus, sodass Sie über SSH eine Verbindung zu Ihrer Entwicklungsumgebung herstellen können:
    - Installiert das [Session Manager-Plugin](#) für AWS CLI
    - Ändert Ihr lokales AWS Config Konto und fügt ein CodeCatalyst Profil hinzu, sodass Sie die SSO-Anmeldung durchführen können. Weitere Informationen finden Sie unter [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#).
    - Ändert Ihre lokale SSH-Konfiguration und fügt die erforderliche Konfiguration für die Verbindung mit Ihrer Entwicklungsumgebung über SSH hinzu.
    - Fügt ein Skript in das `~/ .aws/codecatalyst-dev-env` Verzeichnis ein, das vom SSH-Client verwendet wird, um eine Verbindung zu Ihrer Entwicklungsumgebung herzustellen. Dieses Skript ruft die [CodeCatalyst StartDevEnvironmentSession API](#) auf und verwendet das AWS Systems Manager Session Manager Plugin, um eine AWS Systems Manager Sitzung mit Ihrer Entwicklungsumgebung einzurichten, die vom lokalen SSH-Client verwendet wird, um eine sichere Verbindung zur Remote-Entwicklungsumgebung herzustellen.
  - Melden Sie sich mit dem zweiten Befehl CodeCatalyst mit AWS SSO bei Amazon an. [Dieser Befehl fordert Anmeldeinformationen an und ruft sie ab, sodass das Skript im ~/ .aws/codecatalyst-dev-env Verzeichnis die API aufrufen CodeCatalyst StartDevEnvironmentSession kann.](#) Dieser Befehl sollte jedes Mal ausgeführt

werden, wenn Ihre Anmeldeinformationen ablaufen. Wenn Sie den letzten Befehl im Modal (`ssh<destination>`) ausführen, erhalten Sie eine Fehlermeldung, wenn Ihre Anmeldeinformationen abgelaufen sind oder Sie die SSO-Anmeldung nicht wie in diesem Schritt beschrieben durchgeführt haben.

- Stellen Sie mit dem dritten Befehl über SSH eine Connect zu Ihrer angegebenen Entwicklungsumgebung her. Dieser Befehl hat die folgende Struktur:

```
ssh codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

Sie können diesen Befehl auch verwenden, um andere vom SSH-Client zugelassene Aktionen auszuführen, z. B. die Portweiterleitung oder das Hoch- und Herunterladen von Dateien:

- Portweiterleitung:

```
ssh -L <local-port>:127.0.0.1:<remote-port> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

- Eine Datei in das Home-Verzeichnis in Ihrer Entwicklungsumgebung hochladen:

```
scp -0 </path-to-local-file> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>:</path-to-remote-file-or-directory>
```

## Konfiguration Ihrer Entwicklungsumgebung

Eine Dev-Datei ist ein offener Standard, mit dem Sie Ihre Entwicklungsumgebungen in Ihrem Team individuell anpassen können. Eine Dev-Datei ist eine YAML-Datei, die Ihre benötigten Entwicklungstools kodifiziert. Durch die Konfiguration einer Dev-Datei können Sie vorab festlegen, welche Projekttools und Anwendungsbibliotheken Sie benötigen, und Amazon CodeCatalyst installiert sie für Sie in Ihrer Entwicklungsumgebung. Die Dev-Datei ist spezifisch für das Repository, für das sie erstellt wurde, und Sie können für jedes Repository eine separate Dev-Datei erstellen. Ihre Entwicklungsumgebung unterstützt Befehle und Ereignisse und stellt ein standardmäßiges universelles Devfile-Image bereit.

Wenn Sie ein Projekt mit dem leeren Blueprint erstellen, können Sie eine Dev-Datei manuell erstellen. Wenn Sie ein Projekt mit einem anderen Blueprint erstellen, wird CodeCatalyst automatisch eine Dev-Datei erstellt. Das `/projects` Verzeichnis einer Entwicklungsumgebung speichert die Dateien, die aus dem Quell-Repository und der Dev-Datei abgerufen werden. Das `/home` Verzeichnis, das leer ist, wenn Sie zum ersten Mal eine Entwicklungsumgebung erstellen, speichert

die Dateien, die Sie während der Verwendung Ihrer Entwicklungsumgebung erstellen. Alles in den /home Verzeichnissen /projects und einer Entwicklungsumgebung wird persistent gespeichert.

#### Note

Der /home Ordner ändert sich nur, wenn Sie den Namen der Dev-Datei oder der Devfile-Komponente ändern. Wenn Sie den Namen der Devfile- oder Devfile-Komponente ändern, wird der Inhalt des /home Verzeichnisses ersetzt und Ihre vorherigen /home Verzeichnisdaten können nicht wiederhergestellt werden.

Wenn Sie eine Entwicklungsumgebung mit einem Quell-Repository erstellen, das keine Dev-Datei im Stammverzeichnis enthält, oder wenn Sie eine Entwicklungsumgebung ohne Quell-Repository erstellen, wird automatisch eine standardmäßige universelle Dev-Datei auf das Quell-Repository angewendet. Für alle IDEs wird dasselbe standardmäßige universelle Devfile-Image verwendet. CodeCatalyst unterstützt derzeit die Devfile-Version 2.0.0. Weitere Informationen zur Dev-Datei finden Sie unter [Devfile-Schema — Version 2.0.0](#).

#### Note

Sie können nur öffentliche Container-Images in Ihre Devfile-Datei aufnehmen.

Beachten Sie, dass VPC-verbundene Entwicklungsumgebungen nur die folgenden Devfile-Images unterstützen:

- Universelles Image
- Private Amazon ECR-Images, wenn sich das Repository in derselben Region wie die VPC befindet

#### Themen

- [Bearbeiten einer Repository-Dev-Datei für eine Entwicklungsumgebung in CodeCatalyst](#)
- [Bearbeiten einer Repository-Dev-Datei für eine Entwicklungsumgebung in einer IDE](#)
- [Eine Repository-Dev-Datei für eine Entwicklungsumgebung verschieben](#)
- [Wiederherstellungsmodus](#)
- [Devfile-Funktionen werden unterstützt von CodeCatalyst](#)
- [Beispiel: Konfiguration der Dev-Datei für Ihre Entwicklungsumgebung](#)

- [Devfile-Befehle](#)
- [Devfile-Ereignisse](#)
- [Devfile-Komponenten](#)
- [Universelle Devfile-Images](#)

## Bearbeiten einer Repository-Dev-Datei für eine Entwicklungsumgebung in CodeCatalyst

Um die Repository-Dev-Datei zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, das das Quell-Repository enthält, für das Sie die Dev-Datei bearbeiten möchten.
3. Wählen Sie im Navigationsbereich Code aus.
4. Wählen Sie Source Repositories aus.
5. Wählen Sie das Quell-Repository aus, das die Dev-Datei enthält, die Sie bearbeiten möchten.
6. Wählen Sie die Datei aus der `devfile.yaml` Dateiliste aus.
7. Wählen Sie Bearbeiten aus.
8. Bearbeiten Sie die Dev-Datei.
9. Wählen Sie Commit oder erstellen Sie eine Pull-Anfrage, damit ein Teammitglied die Änderungen überprüfen und genehmigen kann.

### Note

Wenn Sie Ihre Dev-Datei bearbeiten, müssen Sie die Dev-Datei neu starten, damit die Änderungen wirksam werden. Dies kann durch Ausführen geschehen. `/aws/mde/mde start --location devfile.yaml` Wenn beim Starten Ihrer Dev-Datei ein Problem auftritt, wechselt sie in den Wiederherstellungsmodus. Wenn Sie jedoch eine Dev-Datei bearbeiten, die mit einer VPC-verbundenen Entwicklungsumgebung verknüpft ist, müssen Sie stattdessen die Entwicklungsumgebung neu starten, damit die Änderungen wirksam werden.

Sie können überprüfen, welche Dev-Datei verwendet wird, indem Sie den Befehl ausführen. `/aws/mde/mde status` Das Ortsfeld enthält den Pfad der Dev-Datei relativ zum Ordner der `/projects` Umgebung.

```
{
  "status": "STABLE",
  "location": "devfile.yaml"
}
```

## Bearbeiten einer Repository-Dev-Datei für eine Entwicklungsumgebung in einer IDE

Um die Konfiguration einer Entwicklungsumgebung zu ändern, müssen Sie die Dev-Datei bearbeiten. Wir empfehlen, dass Sie die Dev-Datei in einer unterstützten IDE bearbeiten und dann Ihre Entwicklungsumgebung aktualisieren. Sie können die Dev-Datei aber auch vom Stammverzeichnis des Quell-Repositorys aus bearbeiten. CodeCatalyst Wenn Sie die Dev-Datei in einer unterstützten IDE bearbeiten, müssen Sie Ihre Änderungen übertragen und in das Quell-Repository übertragen oder eine Pull-Anfrage erstellen, damit ein Teammitglied die Änderungen an der Devdatei überprüfen und genehmigen kann.

- [Bearbeiten der Repository-Devdatei für eine Entwicklungsumgebung in AWS Cloud9](#)
- [Bearbeiten der Repository-Devdatei für eine Entwicklungsumgebung in VS Code](#)
- [Bearbeiten der Repository-Devdatei für eine Entwicklungsumgebung in JetBrains](#)

## Eine Repository-Dev-Datei für eine Entwicklungsumgebung verschieben

Sie können die Standard-Dev-Datei in `/projects/devfile.yaml` Ihr Quellcode-Repository verschieben. Verwenden Sie den folgenden Befehl, um den Speicherort der Dev-Datei zu aktualisieren: `/aws/mde/mde start --location repository-name/devfile.yaml`

## Wiederherstellungsmodus

Wenn beim Starten Ihrer Dev-Datei ein Problem auftritt, wechselt sie in den Wiederherstellungsmodus, sodass Sie immer noch eine Verbindung zu Ihrer Umgebung herstellen und Ihre Dev-Datei reparieren können. Wenn Sie sich im Wiederherstellungsmodus befinden, enthält die Ausführung `/aws/mde/mde status` nicht den Speicherort Ihrer Dev-Datei.

```
{
```

```
"status": "STABLE"  
}
```

Sie können den Fehler in den Protokollen unter `überprüfen/aws/mde/logs`, die Dev-Datei korrigieren und erneut versuchen, es auszuführen/`aws/mde/mde start`.

## Devfile-Funktionen werden unterstützt von CodeCatalyst

CodeCatalyst unterstützt die folgenden Devfile-Funktionen in Version 2.0.0. Weitere Informationen zur Dev-Datei finden Sie unter [Devfile-Schema — Version 2.0.0](#).

Funktion	Typ
<code>exec</code>	Befehl
<code>postStart</code>	Ereignis
<code>container</code>	Komponente
<code>args</code>	Eigenschaften der Komponenten
<code>env</code>	Eigenschaften der Komponenten
<code>mountSources</code>	Eigenschaften der Komponenten
<code>volumeMounts</code>	Eigenschaften der Komponenten

## Beispiel: Konfiguration der Dev-Datei für Ihre Entwicklungsumgebung

Das Folgende ist ein Beispiel für eine einfache Dev-Datei.

```
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
```

Start-, Befehls- und Ereignisprotokolle von Devfiles werden erfasst und gespeichert in `/aws/mde/logs`. Um das Verhalten von Devfiles zu debuggen, starten Sie Ihre Entwicklungsumgebung mit einer funktionierenden Dev-Datei und greifen Sie auf die Protokolle zu.

## Devfile-Befehle

Unterstützt derzeit CodeCatalyst nur `exec` Befehle in Ihrer Dev-Datei. Weitere Informationen finden Sie unter [Befehle hinzufügen](#) in der Devfile.io-Dokumentation.

Das folgende Beispiel zeigt Ihnen, wie Sie `exec` Befehle in Ihrer Dev-Datei angeben.

```
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"
```

Nachdem Sie mit Ihrer Entwicklungsumgebung verbunden sind, können Sie definierte Befehle über das Terminal ausführen.

```
/aws/mde/mde command <command-id>
/aws/mde/mde command executescript
```

Bei Befehlen mit langer Laufzeit können Sie das Streaming-Flag verwenden `-s`, um die Ausführung des Befehls in Echtzeit auszugeben.

```
/aws/mde/mde -s command <command-id>
```

#### Note

`command-id` muss in Kleinbuchstaben geschrieben werden.

## Exec-Parameter werden unterstützt von CodeCatalyst

CodeCatalyst unterstützt die folgenden exec Parameter in der Devfile-Version 2.0.0.

- `commandLine`
- `component`
- `id`
- `workingDir`

## Devfile-Ereignisse

Unterstützt derzeit CodeCatalyst nur `postStart` Ereignisse in Ihrer Dev-Datei. Weitere Informationen finden Sie [postStartObject](#) in der Devfile.io-Dokumentation.



Das folgende Beispiel zeigt Ihnen, wie Sie Ihrer Dev-Datei `postStart` Event-Bindungen hinzufügen.

```
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
```

Nach dem Start führt Ihre Entwicklungsumgebung die angegebenen `postStart` Befehle in der Reihenfolge aus, in der sie definiert sind. Wenn ein Befehl fehlschlägt, läuft die Entwicklungsumgebung weiter und die Ausführungsausgabe wird in den Protokollen unter `gespeichert/aws/mde/logs`.

## Devfile-Komponenten

Unterstützt derzeit CodeCatalyst nur `container` Komponenten in Ihrer Dev-Datei. Weitere Informationen finden Sie unter [Hinzufügen von Komponenten](#) in der Devfile.io-Dokumentation.

Das folgende Beispiel zeigt Ihnen, wie Sie Ihrem Container in Ihrer Dev-Datei einen Startbefehl hinzufügen.

```
components:
  - name: test
    container:
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      command: ['sleep', 'infinity']
```

### Note

Wenn der Container einen kurzlebigen Eingabebefehl hat, müssen Sie diesen einschließen, `command: ['sleep', 'infinity']` um den Container am Laufen zu halten.

CodeCatalyst unterstützt auch die folgenden Eigenschaften in Ihrer Container-Komponente: `argsenv`, `mountSources`, und `volumeMounts`.

## Universelle Devfile-Images

Das standardmäßige Universal-Image enthält die am häufigsten verwendeten Programmiersprachen und zugehörige Tools, die für Ihre IDE verwendet werden können. Wenn kein Bild angegeben ist, CodeCatalyst stellt es dieses Bild bereit und enthält Tools, die von verwaltet werden CodeCatalyst. Informationen darüber, wie Sie über neue Image-Versionen informiert werden, finden Sie unter [Universelle Bildbenachrichtigungen mit SNS](#).

### Note

Das `public.ecr.aws/aws-mde/universal-image:latest` Bild erhält das `public.ecr.aws/aws-mde/universal-image:3.0` Bild.

Amazon CodeCatalyst unterstützt die folgenden Devfile-Images.

Image-Version	Image-Kennung
Universal image 1.0	<code>public.ecr.aws/aws-mde/universal-image:1.0</code>
Universal image 2.0	<code>public.ecr.aws/aws-mde/universal-image:2.0</code>
Universal image 3.0	<code>public.ecr.aws/aws-mde/universal-image:3.0</code>

### Note

Wenn Sie verwenden AWS Cloud9, funktioniert die automatische Vervollständigung nach dem Upgrade auf nicht für PHP, Ruby und CSS. `universal-image:3.0`

## Themen

- [Universelle Bildbenachrichtigungen mit SNS](#)

- [Runtime-Versionen von Universal Image 1.0](#)
- [Laufzeitversionen von Universal Image 2.0](#)
- [Laufzeitversionen von Universal Image 3.0](#)

## Universelle Bildbenachrichtigungen mit SNS

CodeCatalyst bietet einen universellen Bildbenachrichtigungsdienst. Sie können es verwenden, um ein Amazon Simple Notification Service (SNS) -Thema zu abonnieren, das Sie benachrichtigt, wenn CodeCatalyst Universal-Image-Updates veröffentlicht wurden. Weitere Informationen zu SNS-Themen finden Sie unter [Was ist Amazon Simple Notification Service?](#) .

Immer wenn neue Universal-Images veröffentlicht werden, senden wir Benachrichtigungen an Abonnenten. In diesem Abschnitt wird beschrieben, wie Sie CodeCatalyst Universal-Image-Updates abonnieren können.

### Beispielnachricht

```
{
  "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"
}
```

Um CodeCatalyst universelle Image-Updates über die Amazon SNS SNS-Konsole zu abonnieren

1. Öffnen Sie die Amazon SNS SNS-Konsole zum [Dashboard](#).

2. Wählen Sie in der Navigationsleiste Ihre AWS-Region.
3. Wählen Sie im Navigationsbereich Subscriptions (Abonnements) und dann Create subscription (Abonnement erstellen) aus.
4. Geben Sie im Feld Thema ARN den Text `arn:aws:sns:us-east-1:089793673375:universal-image-updates`.
5. Wählen Sie unter Protocol (Protokoll) die Option Email (E-Mail) aus.
6. Geben Sie in Endpoint eine E-Mail-Adresse ein. Diese E-Mail-Adresse wird für den Empfang von Benachrichtigungen verwendet.
7. Wählen Sie Create subscription (Abonnement erstellen) aus.
8. Sie erhalten eine Bestätigungs-E-Mail mit dem Betreff „AWS Benachrichtigung — Abonnementbestätigung“. Öffnen Sie die E-Mail und wählen Sie Abonnement bestätigen.

Um sich über die Amazon SNS SNS-Konsole von CodeCatalyst universellen Image-Updates abzumelden

1. Öffnen Sie die Amazon SNS SNS-Konsole zum [Dashboard](#).
2. Wählen Sie in der Navigationsleiste Ihre AWS-Region.
3. Wählen Sie im Navigationsbereich Abonnements und dann das Abonnement aus, von dem Sie sich abmelden möchten.
4. Wählen Sie Aktionen und dann Abonnements löschen aus.
5. Wählen Sie Löschen aus.

## Runtime-Versionen von Universal Image 1.0

In der folgenden Tabelle sind die verfügbaren Laufzeiten für `universal-image:1.0` aufgeführt.

### **universal-image:1.0** Laufzeitversionen

Laufzeitname	Version	Spezifische Haupt- und neueste Unterversion
AWS-CLI	2.11	<code>aws-cli: 2.x</code>
Docker Compose	2.16	<code>docker-compose: 2.x</code>
dotnet	6.0	<code>dotnet: 6.x</code>

Laufzeitname	Version	Spezifische Haupt- und neueste Unterversion
	7.0	dotnet: 7.x
Golang	1,19	golang: 1.x
Java	corretto11	java: corretto11.x
	Korretto 17	java: corretto17.x
nodejs	14,20	nodejs: 14.x
	16,19	nodejs: 16.x
OpenSSL	1,0	openssl: 1.x
	1.1	
php	7.2	php: 7.x
python	3.9	python: 3.x
	3,10	
ruby	3.1	ruby: 3.x
Terraform	1.4	terraform: 1.x

## Laufzeitversionen von Universal Image 2.0

In der folgenden Tabelle sind die verfügbaren Laufzeiten für `universal-image:2.0` aufgeführt.

### **universal-image:2.0** Laufzeitversionen

Laufzeitname	Version	Spezifische Haupt- und neueste Unterversion
AWS-CLI	2.11	aws-cli: 2.x
Docker Compose	2.17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x

Laufzeitname	Version	Spezifische Haupt- und neueste Unterversion
	7.0	dotnet: 7.x
Golang	1,20	golang: 1.x
Java	corretto11	java: corretto11.x
	Korretto 17	java: corretto17.x
nodejs	16,19	nodejs: 16.x
OpenSSL	1,0	openssl: 1.x
	1.1	
php	7.2	php: 7.x
python	3.9	python: 3.x
	3,10	
ruby	3.2	ruby: 3.x
Terraform	1.4	terraform: 1.x

## Laufzeitversionen von Universal Image 3.0

In der folgenden Tabelle sind die verfügbaren Laufzeiten für `universal-image:3.0` aufgeführt.

### **universal-image:3.0** Laufzeitversionen

Laufzeitname	Version	Spezifische Haupt- und neueste Unterversion
AWS-CLI	2.11	aws-cli: 2.x
Docker Compose	2.17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x

Laufzeitname	Version	Spezifische Haupt- und neueste Unterversion
Golang	1,21	golang: 1.x
Java	corretto11	java: corretto11.x
	Korretto 17	java: corretto17.x
nodejs	18,17	nodejs: 18.x
	20.6	nodejs: 20.x
OpenSSL	3.0	openssl: 3.x
php	8.2	php: 8.x
python	3.9	python: 3.x
	3,11	
ruby	3.2	ruby: 3.x
Terraform	1.5	terraform: 1.x

## Verwenden von Entwicklungsumgebungen mit einer VPC-Verbindung

Eine VPC-Verbindung ist eine CodeCatalyst Ressource, die alle Konfigurationen enthält, die Ihr Workflow für den Zugriff auf eine VPC benötigt. Space-Administratoren können im Namen von Space-Mitgliedern ihre eigenen VPC-Verbindungen in der CodeCatalyst Amazon-Konsole hinzufügen. Durch Hinzufügen einer VPC-Verbindung können Space-Mitglieder Workflow-Aktionen ausführen und Entwicklungsumgebungen erstellen, die den Netzwerkregeln entsprechen und auf Ressourcen in der zugehörigen VPC zugreifen können.

Sie können einer VPC-Verbindung nur bei der Erstellung der Entwicklungsumgebung eine Entwicklungsumgebung zuordnen. Sie können die VPC-Verbindung, die Ihrer Entwicklungsumgebung zugeordnet ist, nicht ändern, nachdem Sie sie erstellt haben. Wenn Sie eine andere VPC-Verbindung verwenden möchten, müssen Sie Ihre aktuelle Entwicklungsumgebung löschen und eine neue erstellen.

**⚠ Important**

Entwicklungsumgebungen mit einer VPC-Verbindung unterstützen keine [verlinkten Quell-Repositorys von Drittanbietern](#). CodeCatalyst

Beachten Sie, dass Entwicklungsumgebungen bei der Erstellung mehrere AWS Ressourcen und Dienste verwenden. Das bedeutet, dass Entwicklungsumgebungen eine Verbindung zu den folgenden AWS Diensten herstellen:

- Amazon CodeCatalyst
- AWS SSM
- AWS KMS
- Amazon ECR
- Amazon CloudWatch
- Amazon ECS

**ℹ Note**

AWS Toolkit unterstützt die Erstellung von Entwicklungsumgebungen mit einer zugehörigen VPC-Verbindung nicht. Beachten Sie auch, dass es bei Verwendung einer anderen IDE als AWS Cloud9 zu Ladezeiten von etwa fünf Minuten kommen kann.

Sie benötigen die Rolle Space-Administrator oder Poweruser, um VPC-Verbindungen auf Space-Ebene verwalten zu können. Weitere Informationen zu VPCs finden Sie unter [Verwaltung von Amazon VPCs CodeCatalyst im CodeCatalyst Administratorhandbuch](#).

## Verwenden von Entwicklungsumgebungen mit einer IDE

Sie können Dev Environments verwenden, um schnell an dem Code zu arbeiten, der in den Quell-Repositorys Ihres Projekts gespeichert ist. Entwicklungsumgebungen erhöhen Ihre Entwicklungsgeschwindigkeit, da Sie sofort mit dem Programmieren in einer projektspezifischen, voll funktionsfähigen Cloud-Entwicklungsumgebung mit einer unterstützten integrierten Entwicklungsumgebung (IDE) beginnen können.



Informationen zum Arbeiten mit CodeCatalyst einer IDE finden Sie in der folgenden Dokumentation.

- [Amazon CodeCatalyst für JetBrains IDEs](#)
- [Amazon CodeCatalyst für VS Code](#)
- [Amazon CodeCatalyst für AWS Cloud9](#)

## Kontingente für Entwicklungsumgebungen in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Limits für Entwicklungsumgebungen in Amazon CodeCatalyst beschrieben. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

Anzahl der Stunden der Entwicklungsumgebung pro Monat	Die Stunden der Entwicklungsumgebung hängen von den allgemeinen Speicherlimits für Ihren Speicherplatz ab. Weitere Informationen finden Sie unter <a href="#">-Preise</a> und <a href="#">Behebung von Problemen mit Entwicklungsumgebungen</a> .
Menge an Speicherplatz in der Entwicklungsumgebung pro Speicherplatz	Der Speicherplatz in der Entwicklungsumgebung wird von den allgemeinen Speicherlimits für Ihren Speicherplatz beeinflusst. Weitere Informationen finden Sie unter <a href="#">-Preise</a> und <a href="#">Behebung von Problemen mit Entwicklungsumgebungen</a> .
Menge der Rechenleistung in der Entwicklungsumgebung	Die Rechenleistung in der Entwicklungsumgebung wird durch die allgemeinen Speicherlimits für Ihren Speicherplatz beeinflusst. Weitere Informationen finden Sie unter <a href="#">-Preise</a> und <a href="#">Behebung von Problemen mit Entwicklungsumgebungen</a> .

# Pakete in CodeCatalyst

Amazon bietet CodeCatalyst einen vollständig verwalteten Paket-Repository-Service, der es Ihrem Entwicklungsteam erleichtert, Softwarepakete, die für die Anwendungsentwicklung verwendet werden, sicher zu speichern und gemeinsam zu nutzen. Diese Pakete werden in Paket-Repositorys gespeichert, die innerhalb von Projekten in CodeCatalyst erstellt und organisiert werden.

CodeCatalyst unterstützt die folgenden Paketformate:

- npm

Pakete in einem Paket-Repository können entdeckt und von Mitgliedern des Projekts, das das Repository enthält, gemeinsam genutzt werden.

Um Pakete zu einem Repository hinzuzufügen, konfigurieren Sie einen Paketmanager so, dass er den Repository-Endpunkt (URL) verwendet. Anschließend können Sie den Paketmanager verwenden, um Pakete im Repository zu veröffentlichen.

Sie können CodeCatalyst Workflows konfigurieren, um Pakete in Paket-Repositorys zu veröffentlichen und Pakete aus CodeCatalyst Paket-Repositorys zu konsumieren. Weitere Informationen zur Verwendung von Paketen in Workflows finden Sie unter [Mit Paketen arbeiten](#).

Sie können Pakete in einem Paket-Repository für ein anderes Repository im selben Projekt verfügbar machen, indem Sie es als Upstream-Repository hinzufügen. Alle Paketversionen, die für das Upstream-Repository verfügbar sind, sind auch für das Downstream-Repository verfügbar.

Sie können Open-Source-Pakete für Ihr CodeCatalyst Repository verfügbar machen, indem Sie öffentliche, externe Repositorien damit verbinden. Weitere Informationen zu Upstream-Repositorys und zur Verbindung mit externen Repositorien, einschließlich einer Liste der unterstützten Repositorien, finden Sie unter [Mit Upstream-Repositorys arbeiten](#)

## Themen

- [Konzepte für Pakete](#)
- [Mit Paket-Repositorys arbeiten](#)
- [Mit Upstream-Repositorys arbeiten](#)
- [Verbindung zu öffentlichen externen Repositorien herstellen](#)
- [Mit Paketen arbeiten](#)

- [Verwenden von npm](#)
- [Kontingente für Pakete](#)

## Konzepte für Pakete

Im Folgenden finden Sie einige Konzepte und Begriffe, die Sie bei der Verwaltung, Veröffentlichung oder Nutzung von Paketen kennen sollten CodeCatalyst.

### Pakete

Ein Paket ist ein Paket, das sowohl Software als auch die Metadaten enthält, die für die Installation der Software und die Auflösung aller Abhängigkeiten erforderlich sind. CodeCatalyst unterstützt das npm-Paketformat.

Ein Paket besteht aus:

- Ein Name (webpackist zum Beispiel der Name eines beliebigen npm-Pakets)
- Ein optionaler [Namespace](#) (zum Beispiel in) `@types @types/node`
- Eine Reihe von [Versionen](#) (zum Beispiel, `1.0.0 1.0.1, 1.0.2`)
- Metadaten auf Paketebene (z. B. npm dist-Tags)

### Paket-Namespaces

Einige Paketformate unterstützen hierarchische Paketnamen, um Pakete in logische Gruppen zu organisieren und Namenskollisionen zu vermeiden. Pakete mit demselben Namen können in verschiedenen Namespaces gespeichert werden. Beispielsweise unterstützt npm Bereiche, und das npm-Paket `@types/node` hat den Bereich und den Namen `@types node`. Es gibt viele andere Paketnamen im Gültigkeitsbereich `@types CodeCatalystIn` wird der Bereich („Typen“) als Paket-Namespace und der Name („Knoten“) als Paketname bezeichnet. Wenn Sie keine Möglichkeit haben, Paketnamen zu gruppieren, kann es schwieriger sein, Namenskollisionen zu vermeiden.

### Versionen von Paketen

Eine Paketversion identifiziert die spezifische Version eines Pakets, z. `@types/node@12.6.9`. Das Format und die Semantik der Versionsnummer variieren je nach Paketformat. Beispielsweise müssen npm-Paketversionen der [Semantic](#) Versioning-Spezifikation entsprechen. In CodeCatalyst

besteht eine Paketversion aus der Versionskennung, package-version-level Metadaten und einer Reihe von Assets.

## Objekte

Ein Asset ist eine einzelne Datei CodeCatalyst , die in einer Paketversion gespeichert ist, z. B. einer .tgz NPM-Datei.

## Paket-Repositoryys

Ein CodeCatalyst Paket-Repository enthält eine Reihe von [Paketen](#), die [Paketversionen](#) enthalten, von denen jede einer Reihe von [Assets](#) zugeordnet ist. Jedes Paket-Repository bietet Endpunkte zum Abrufen und Veröffentlichen von Paketen mithilfe von Tools wie der CLI von Node.js (npm). Sie können in jedem Bereich bis zu 1.000 Paket-Repositoryys erstellen.

Sie können ein Paket-Repository mit einem anderen Repository verknüpfen, indem Sie Upstream-Repositoryys verwenden. Wenn Sie ein Paket-Repository als Upstream-Repository verknüpfen, können Sie die Pakete im verknüpften Repository über das konfigurierte Repository verwenden. Weitere Informationen finden Sie unter [Upstream-Repositoryen](#).

Gateway-Repositoryys sind eine spezielle Art von Paket-Repositoryys, die Pakete von offiziellen externen Paketbehörden abrufen und speichern. Weitere Informationen finden Sie unter [Gateway-Repositoryys](#).

## Gateway-Repositoryys

Ein Gateway-Repository ist eine spezielle Art von Paket-Repository, das mit einer unterstützten externen, offiziellen Paketautorität verbunden ist. Wenn Sie ein Gateway-Repository als [Upstream-Repository](#) hinzufügen, können Sie Pakete von der entsprechenden offiziellen Paketbehörde verwenden. Ihr Downstream-Repository kommuniziert nicht mit dem öffentlichen Repository, sondern alles wird vom Gateway-Repository vermittelt. Auf diese Weise verbrauchte Pakete werden sowohl im Gateway-Repository als auch im Downstream-Repository gespeichert, das die ursprüngliche Anfrage erhalten hat.

Gateway-Repositoryys sind vordefiniert, müssen aber in jedem Projekt erstellt werden, damit sie verwendet werden können. Die folgende Liste enthält alle Gateway-Repositoryys, in denen sie erstellt werden können, CodeCatalyst sowie die Package Authority, mit der sie verbunden sind.

- npm-public-registry-gateway stellt npm-Pakete von npmjs.com bereit.

## Upstream-Repositoryn

Sie können CodeCatalyst verwenden, um eine Upstream-Beziehung zwischen zwei Paket-Repositoryn herzustellen. Ein Paket-Repository ist ein Upstream-Repository eines anderen, wenn auf die darin enthaltenen Paketversionen vom Paket-Repository-Endpunkt des Downstream-Repositorys aus zugegriffen werden kann. Bei einer Upstream-Beziehung werden die Inhalte der beiden Paket-Repositoryn aus Sicht eines Kunden effektiv zusammengeführt.

Wenn ein Paketmanager beispielsweise eine Paketversion anfordert, die in einem Repository nicht existiert, durchsucht er CodeCatalyst dann konfigurierte Upstream-Repositorys nach der Paketversion. Upstream-Repositorys werden in der Reihenfolge durchsucht, in der sie konfiguriert wurden. Sobald ein Paket gefunden wurde, wird die Suche beendet.

## Mit Paket-Repositorys arbeiten

In CodeCatalyst werden Pakete in Paket-Repositorys gespeichert und verwaltet. Um Pakete in einem CodeCatalyst (CodeCatalyst oder einem unterstützten öffentlichen Paket-Repository) zu veröffentlichen oder Pakete daraus zu konsumieren, müssen Sie ein Paket-Repository erstellen und Ihren Paketmanager damit verbinden.

### Themen

- [Ein Paket-Repository erstellen](#)
- [Verbindung zu einem Paket-Repository herstellen](#)
- [Bearbeiten eines Paket-Repositorys](#)
- [Löschen eines Paket-Repositorys](#)

## Ein Paket-Repository erstellen

Führen Sie die folgenden Schritte aus, um ein Paket-Repository in zu erstellen CodeCatalyst.

Um ein Paket-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie ein Paket-Repository erstellen möchten.
3. Wählen Sie im Navigationsbereich Pakete aus.
4. Wählen Sie auf der Seite Paket-Repositorys die Option Paket-Repository erstellen aus.

5. Fügen Sie im Abschnitt Paket-Repository-Details Folgendes hinzu:
  - a. Name des Repositorys. Erwägen Sie die Verwendung eines aussagekräftigen Namens mit Details wie Ihrem Projekt- oder Teamnamen oder der Art und Weise, wie das Repository verwendet werden soll.
  - b. (Optional) Beschreibung des Repositorys. Eine Repository-Beschreibung ist besonders hilfreich, wenn Sie mehrere Repositorys für mehrere Teams in einem Projekt haben.
6. Fügen Sie im Abschnitt Upstream-Repositorys bearbeiten alle Paket-Repositorys hinzu, auf die Sie über Ihr CodeCatalyst Paket-Repository zugreifen möchten. Sie können Gateway-Repositorys hinzufügen, um eine Verbindung zu externen Paket-Repositorys oder anderen Paket-Repositorys herzustellen. CodeCatalyst
  - Wenn ein Paket aus einem Paket-Repository angefordert wird, werden die Upstream-Repositorys in der Reihenfolge durchsucht, in der sie in dieser Liste erscheinen. Sobald ein Paket gefunden wurde, CodeCatalyst wird die Suche beendet. Um die Reihenfolge der Upstream-Repositorys zu ändern, können Sie die Repositorys per Drag-and-Drop in die Liste ziehen oder die Schaltflächen „Neu anordnen“ verwenden.
7. Wählen Sie Erstellen, um Ihr Paket-Repository zu erstellen.

## Verbindung zu einem Paket-Repository herstellen

Um Pakete zu veröffentlichen CodeCatalyst oder von dort aus zu konsumieren CodeCatalyst, müssen Sie Ihren Paketmanager mit den Endpunktinformationen und CodeCatalyst Anmeldeinformationen Ihres Paket-Repositorys konfigurieren. Wenn Sie noch kein Repository erstellt haben, können Sie dies tun, indem Sie den Anweisungen unter folgen [Ein Paket-Repository erstellen](#).

Anweisungen, wie Sie den npm-Paketmanager mit einem CodeCatalyst Paket-Repository verbinden, finden Sie unter [Konfiguration und Verwendung von npm](#).

## Bearbeiten eines Paket-Repositorys

Gehen Sie wie folgt vor, um die Beschreibung eines Paket-Repositorys und die Upstream-Repositorys zu bearbeiten.

Um ein Paket-Repository zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, das das Paket-Repository enthält, das Sie bearbeiten möchten.

3. Wählen Sie im Navigationsbereich Pakete aus.
4. Wählen Sie auf der Seite Paket-Repositoryys das Repository aus, das Sie löschen möchten.
5. Wählen Sie das Drop-down-Menü „Aktionen“ und dann „Bearbeiten“.
6. Bearbeiten Sie die Repository-Beschreibung und die Upstream-Repositoryys. Weitere Hinweise zu Upstream-Repositoryys finden Sie unter. [Mit Upstream-Repositoryys arbeiten](#)
7. Wählen Sie Speichern.

## Löschen eines Paket-Repositoryys

Gehen Sie wie folgt vor, um ein Paket-Repository in zu löschen CodeCatalyst.

Um ein Paket-Repository zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, das das Paket-Repository enthält, das Sie löschen möchten.
3. Wählen Sie im Navigationsbereich Pakete aus.
4. Wählen Sie auf der Seite Paket-Repositoryys das Repository aus, das Sie löschen möchten.
5. Wählen Sie das Drop-down-Menü „Aktionen“ und dann „Löschen“.
6. Lesen Sie die bereitgestellten Informationen zu den Auswirkungen des Löschens eines Paket-Repositoryys.
7. Geben Sie es delete in das Eingabefeld ein und wählen Sie Löschen.

## Mit Upstream-Repositoryys arbeiten

Sie können sowohl Gateway-Repositoryys als auch andere Paket-Repositoryys als Upstreams mit Ihren Paket-Repositoryys verbinden. Dadurch kann ein Paketmanager-Client über einen einzigen Paket-Repository-Endpunkt auf die Pakete zugreifen, die in mehr als einem Paket-Repository enthalten sind. Im Folgenden sind die Hauptvorteile der Verwendung von Upstream-Repositoryys aufgeführt:

- Sie müssen Ihren Paketmanager nur mit einem einzigen Repository-Endpunkt konfigurieren, um Daten aus mehreren Quellen abrufen zu können.
- Pakete, die aus einem Upstream-Repository abgerufen werden, werden in Ihrem Downstream-Repository gespeichert. Dadurch wird sichergestellt, dass Ihre Pakete auch dann verfügbar sind, wenn es im Upstream-Repository zu unerwarteten Ausfällen kommt.

Sie können Upstream-Repositoryys hinzufügen, wenn Sie ein Paket-Repository erstellen. Sie können in der Konsole auch Upstream-Repositoryys zu vorhandenen Paket-Repositoryys hinzufügen oder daraus entfernen. CodeCatalyst

Wenn Sie ein Gateway-Repository als Upstream-Repository hinzufügen, wird das Paket-Repository mit dem entsprechenden öffentlichen Paket-Repository des Gateway-Repositoryys verbunden. Eine Liste der unterstützten öffentlichen Paket-Repositoryys finden Sie unter [Unterstützte externe Paket-Repositoryys und ihre Gateway-Repositoryys](#).

Sie können mehrere Repositoryys als Upstream-Repositoryys miteinander verknüpfen. Nehmen wir zum Beispiel an, dass Ihr Team ein Repository mit dem Namen erstellt `project-repo` und bereits ein anderes Repository mit dem Namen verwendet `team-repo`, dem das als Upstream-Repository `npm-public-registry-gateway` hinzugefügt wurde, das mit dem öffentlichen NPM-Repository verbunden ist. `npmjs.com` Sie können es `team-repo` als Upstream-Repository zu `project-repo` hinzufügen. In diesem Fall müssen Sie nur Ihren Paketmanager so konfigurieren, dass er Pakete aus `project-repo`, `team-repo` `npm-public-registry-gateway`, und abrufen kann `npmjs.com`. `project-repo`

## Themen

- [Hinzufügen eines Upstream-Repositoryys](#)
- [Die Suchreihenfolge von Upstream-Repositoryys bearbeiten](#)
- [Eine Paketversion mit Upstream-Repositoryys anfordern](#)
- [Entfernen eines Upstream-Repositoryys](#)

## Hinzufügen eines Upstream-Repositoryys

Wenn Sie ein öffentliches Paket-Repository oder ein anderes CodeCatalyst Paket-Repository als Upstream-Repository zu Ihrem Downstream-Repository hinzufügen, werden alle Pakete im Upstream-Repository für Paketmanager verfügbar, die mit dem Downstream-Repository verbunden sind.

Um ein Upstream-Repository hinzuzufügen

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie auf der Seite Paket-Repositoryys das Paket-Repository aus, dem Sie ein Upstream-Repository hinzufügen möchten.
3. Wählen Sie das Drop-down-Menü „Aktionen“ und dann „Bearbeiten“.



4. Wählen Sie im Abschnitt Upstream-Repositorys die Option Upstream-Repository hinzufügen aus.
5. Wählen Sie die Suchleiste, um eine Liste der verfügbaren Repositorys aufzurufen und zu durchsuchen. Sie können unterstützte öffentliche Paket-Repositorys oder andere Repositorys als CodeCatalyst Upstream-Repositorys hinzufügen. Wenn Sie das Repository gefunden haben, das Sie hinzufügen möchten, wählen Sie es aus der Liste aus.

#### Note

In Gateway-Repositorys gibt es ein `npm-public-registry-gatewayRepository`. Um eine Verbindung zu öffentlichen externen Paketbehörden wie `npmjs.com` herzustellen, wird ein Gateway-Repository als Zwischenrepositorien CodeCatalyst verwendet, die Pakete suchen und speichern, die aus externen Repositorys abgerufen wurden. Dies spart Zeit und Datenübertragung, da alle Paket-Repositorys in einem Projekt Pakete aus dem Gateway-Repository verwenden.

6. Wenn Sie alle Repositorys ausgewählt haben, die Sie als Upstream-Repositorys hinzufügen möchten, wählen Sie Hinzufügen.
7. Weitere Informationen zum Ändern der Suchreihenfolge von Upstream-Repositorys finden Sie unter [Die Suchreihenfolge von Upstream-Repositorys bearbeiten](#)

Wenn Sie ein Upstream-Repository hinzugefügt haben, können Sie einen Paketmanager verwenden, der mit Ihrem lokalen Repository verbunden ist, um Pakete aus dem Upstream-Repository abzurufen. Sie müssen Ihre Paketmanager-Konfiguration nicht aktualisieren. Weitere Hinweise zum Anfordern von Paketversionen aus einem Upstream-Repository finden Sie unter [Eine Paketversion mit Upstream-Repositorys anfordern](#).

## Die Suchreihenfolge von Upstream-Repositorys bearbeiten

CodeCatalyst durchsucht Upstream-Repositorys in ihrer konfigurierten Suchreihenfolge. Wenn ein Paket gefunden wird, wird die Suche CodeCatalyst beendet. Sie können die Reihenfolge ändern, in der die Upstream-Repositorys nach Paketen durchsucht werden.

Um die Suchreihenfolge der Upstream-Repositorys zu bearbeiten

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie auf der Seite Repositorys das Paket-Repository aus, dessen Suchreihenfolge im Upstream-Repository Sie bearbeiten möchten.

3. Wählen Sie das Drop-down-Menü „Aktionen“ und dann „Bearbeiten“.
4. Im Abschnitt Upstream-Repositoryys können Sie die Upstream-Repositoryys und ihre Suchreihenfolge einsehen. Um die Suchreihenfolge zu ändern, ziehen Sie die Repositoryys per Drag-and-Drop in die Liste oder verwenden Sie die Schaltflächen „Neu anordnen“.
5. Wenn Sie mit der Bearbeitung der Suchreihenfolge der Upstream-Repositoryys fertig sind, wählen Sie Speichern.

## Eine Paketversion mit Upstream-Repositoryys anfordern

Das folgende Beispiel zeigt die möglichen Szenarien, wenn ein Paketmanager ein Paket aus einem CodeCatalyst Paket-Repository anfordert, das über Upstream-Repositoryys verfügt.

In diesem Beispiel fordert ein Paketmanager beispielsweise eine Paketversion von einem Paket-Repository mit dem Namen `annpm`, `downstream` das mehrere Upstream-Repositoryys hat. Wenn das Paket angefordert wird, kann Folgendes passieren:

- Wenn es die angeforderte Paketversion `downstream` enthält, wird es an den Client zurückgegeben.
- Wenn `downstream` es die angeforderte Paketversion nicht enthält, CodeCatalyst wird in `downstream` den Upstream-Repositoryys in der konfigurierten Suchreihenfolge danach gesucht. Wenn die Paketversion gefunden wird, wird ein Verweis darauf `downstream` kopiert und die Paketversion wird an den Client zurückgegeben.
- Wenn keines `downstream` der Upstream-Repositoryys die Paketversion enthält, wird eine Not Found HTTP-404-Antwort an den Client zurückgegeben.

Die maximale Anzahl von direkten Upstream-Repositoryys, die für ein Repository zulässig sind, ist 10. Die maximale Anzahl von Repositoryys, in denen CodeCatalyst gesucht wird, wenn eine Paketversion angefordert wird, ist 25.

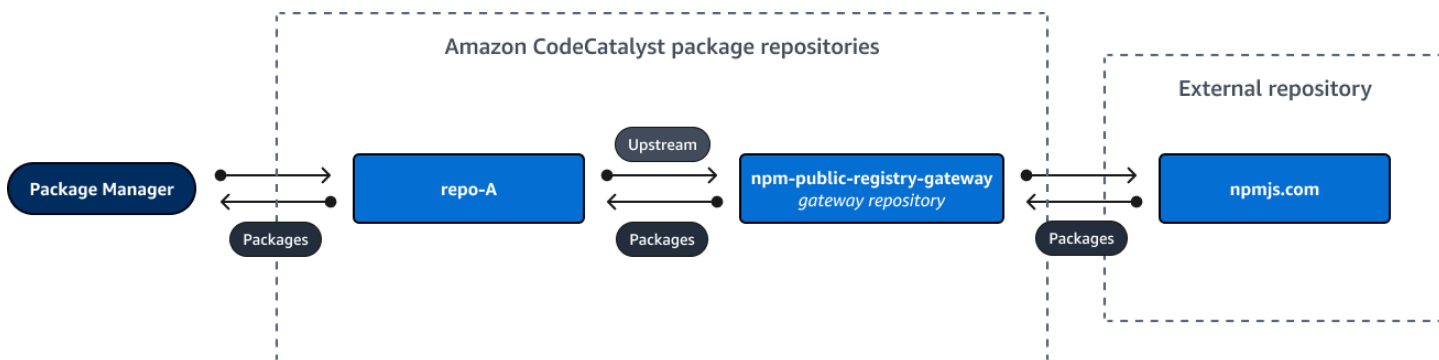
## Aufbewahrung von Paketen aus Upstream-Repositoryys

Wenn eine angeforderte Paketversion in einem Upstream-Repository gefunden wird, wird ein Verweis darauf beibehalten und ist immer in dem Repository verfügbar, das sie angefordert hat. Dadurch wird sichergestellt, dass Sie Zugriff auf Ihre Pakete haben, falls es zu einem unerwarteten Ausfall des Upstream-Repositoryys kommt. Die beibehaltene Paketversion ist von keinem der folgenden Faktoren betroffen:

- Das Upstream-Repository wird gelöscht.
- Trennen des Upstream-Repositorys vom Downstream-Repository.
- Löschen der Paketversion aus dem Upstream-Repository.
- Bearbeiten der Paketversion im Upstream-Repository (z. B. durch Hinzufügen eines neuen Assets).

## Pakete über eine Upstream-Beziehung abrufen

CodeCatalyst kann Pakete über mehrere verknüpfte Repositories abrufen, die als Upstream-Repositories bezeichnet werden. Wenn ein CodeCatalyst Paket-Repository eine Upstream-Verbindung zu einem anderen CodeCatalyst Paket-Repository hat, das eine Upstream-Verbindung zu einem Gateway-Repository hat, werden Anfragen für Pakete, die sich nicht im Upstream-Repository befinden, aus dem externen Repository kopiert. Stellen Sie sich zum Beispiel die folgende Konfiguration vor: Ein Repository mit dem Namen `repo-A` hat eine Upstream-Verbindung zum Gateway-Repository, `npm-public-registry-gateway`. `npm-public-registry-gateway` hat eine Upstream-Verbindung zum öffentlichen Paket-Repository, <https://npmjs.com>.



Wenn npm es für die Verwendung des `repo-A` Repositories konfiguriert ist, `npm install` initiiert das Ausführen das Kopieren von Paketen von <https://npmjs.com> in `npm-public-registry-gateway`. Die installierten Versionen werden ebenfalls abgerufen. `repo-A` Das folgende Beispiel wird installiert `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
```

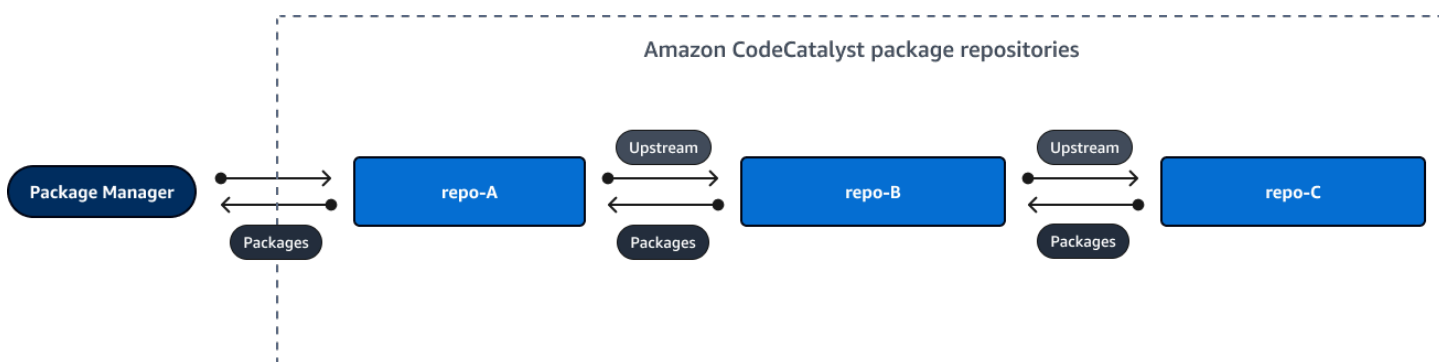
`repo-A` Enthält nach der Ausführung `npm install` nur die neueste Version (`lodash 4.17.20`), da dies die Version ist, die npm von `repo-A` abgerufen wurde.

Da es `npm-public-registry-gateway` eine externe Upstream-Verbindung zu <https://npmjs.com> gibt, werden alle Paketversionen, die von <https://npmjs.com> importiert wurden, in `npm-public-registry-gateway` gespeichert. Diese Paketversionen könnten von jedem Downstream-Repository mit einer Upstream-Verbindung abgerufen worden sein `npm-public-registry-gateway`.

Der Inhalt von `npm-public-registry-gateway` bietet Ihnen die Möglichkeit, alle Pakete und Paketversionen zu sehen, die im Laufe der Zeit von <https://npmjs.com> importiert wurden.

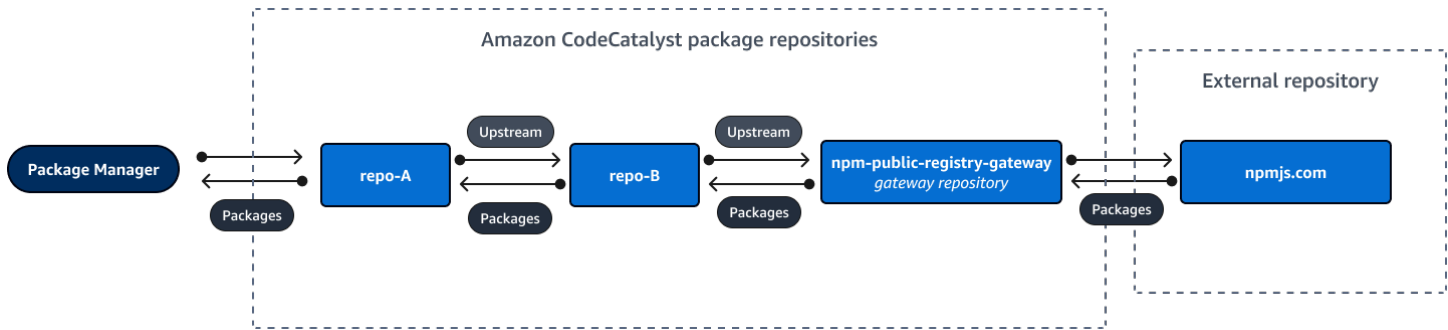
## Aufbewahrung von Paketen in Zwischenrepositorien

CodeCatalyst ermöglicht es Ihnen, Upstream-Repositorys zu verketteten. `repo-A` kann beispielsweise `repo-B` als Upstream-Repository und `repo-B` als Upstream-Repository verwendet werden. `repo-C` Diese Konfiguration macht die Paketversionen in `repo-B` und `repo-C` verfügbar von `repo-A`.



Wenn ein Paketmanager eine Verbindung zum Repository herstellt `repo-A` und eine Paketversion aus dem Repository abrufen `repo-C`, wird die Paketversion nicht im Repository gespeichert. `repo-B` Die Paketversion wird nur im am weitesten nachgeschalteten Repository gespeichert, was in diesem Beispiel der Fall ist. `repo-A` Sie wird in keinen Zwischenrepositorien aufbewahrt. Dies gilt auch für längere Ketten. Wenn es beispielsweise vier Repositorien gäbe: `repo-A`, und `repo-B`, `repo-C` und ein Paketmanager `repo-D`, der verbunden ist, um eine Paketversion `repo-A` abzurufen `repo-D`, würde die Paketversion in oder gespeichert, `repo-A` aber nicht in oder. `repo-B` `repo-C`

Das Verhalten bei der Paketaufbewahrung ist ähnlich, wenn eine Paketversion aus einem öffentlichen Paket-Repository abgerufen wird, mit der Ausnahme, dass die Paketversion immer im Gateway-Repository aufbewahrt wird, das die direkte Upstream-Verbindung zum öffentlichen Repository hat. `repo-A` Hat `repo-B` zum Beispiel ein Upstream-Repository. `repo-B` hat `npm-public-registry-gateway` als Upstream-Repository, das über eine Upstream-Verbindung zum öffentlichen Repository `npmjs.com` verfügt; siehe Abbildung unten.



Wenn ein Paketmanager, der eine Verbindung herstellt, eine bestimmte Paketversion **repo-A** anfordert, zum Beispiel `Lodash 4.17.20`, und die Paketversion in keinem der drei Repositorys vorhanden ist, wird sie von `npmjs.com` abgerufen. Wenn `Lodash 4.17.20` abgerufen wird, wird es beibehalten, da es das am weitesten nachgeschaltete Repository ist und **repo-A npm-public-registry-gateway** da es die Upstream-Verbindung zum öffentlichen externen Repository `npmjs.com` hat. `Lodash 4.17.20` wird nicht beibehalten, da es sich um ein Zwischenarchiv handelt. `repo-B`

## Entfernen eines Upstream-Repositorys

Wenn Sie nicht mehr auf die Pakete in einem Upstream-Repository zugreifen möchten, können Sie das Upstream-Repository aus einem Paket-Repository entfernen.

### Warning

Wenn Sie ein Upstream-Repository entfernen, könnten Sie Upstream-Beziehungsketten unterbrechen, was wiederum Ihre Projekte oder Builds zerstören könnte.

Um ein Upstream-Repository zu entfernen

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie auf der Seite Paket-Repositorys das Paket-Repository aus, aus dem Sie ein Upstream-Repository entfernen möchten.
3. Wählen Sie das Drop-down-Menü „Aktionen“ und dann „Bearbeiten“.
4. Suchen Sie im Abschnitt Upstream-Repositorys das Upstream-Repository, das Sie entfernen möchten, und wählen Sie Entfernen aus.
5. Wenn Sie mit dem Entfernen der Upstream-Repositorys fertig sind, wählen Sie Speichern.

## Verbindung zu öffentlichen externen Repositorien herstellen

Sie können CodeCatalyst Paket-Repositorys mit unterstützten öffentlichen, externen Repositorys verbinden, indem Sie das entsprechende Gateway-Repository als Upstream-Repository hinzufügen. Gateway-Repositorys fungieren als Zwischenrepositorien, die Pakete suchen und speichern, die aus externen Repositorys abgerufen wurden. Dies spart Zeit und Datenübertragung, da alle Paket-Repositorys in einem Projekt Pakete aus dem Gateway-Repository verwenden.

Um mithilfe von Gateway-Repositorys eine Verbindung zu einem öffentlichen Repository herzustellen

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Auf der Seite Paket-Repositorys unter Gateway-Repositorys können Sie eine Liste der unterstützten Gateway-Repositorys und deren Beschreibungen einsehen. Um ein Gateway-Repository verwenden zu können, müssen Sie es zunächst erstellen. Wenn das Gateway-Repository erstellt wurde, werden Datum und Uhrzeit der Erstellung angezeigt. Ist dies nicht der Fall, wählen Sie Create, um es zu erstellen.
3. Wählen Sie das Paket-Repository aus, das Sie mit dem öffentlichen Repository verbinden möchten.
4. Wählen Sie das Drop-down-Menü „Aktionen“ und dann „Bearbeiten“.
5. Um eine Verbindung zum öffentlichen Repository herzustellen, fügen Sie das Gateway-Repository hinzu, das dem öffentlichen Repository entspricht, zu dem Sie eine Verbindung herstellen möchten, als Upstream-Repository.

Wählen Sie im Abschnitt Upstream-Repositorys bearbeiten die Option CodeCatalyst Repository hinzufügen aus.

6. Im Abschnitt Gateway-Repositorys sind alle verfügbaren Gateway-Repositorys aufgeführt. Wenn Sie das Gateway-Repository gefunden haben, das dem öffentlichen, externen Repository entspricht, zu dem Sie eine Verbindung herstellen möchten, wählen Sie es aus der Liste aus und klicken Sie auf Hinzufügen.
7. Wenn ein Paket aus einem Repository angefordert wird, durchsucht CodeCatalyst es die Upstream-Repositorys in der Reihenfolge, in der sie in der Liste Upstream-Repositorys bearbeiten erscheinen. Wenn ein Paket gefunden wird, wird die Suche CodeCatalyst beendet. Um die Reihenfolge der Upstream-Repositorys zu ändern, ziehen Sie die Repositorys per Drag-and-Drop in die Liste oder verwenden Sie die Pfeile zur Neuordnung.
8. Wenn Sie mit dem Hinzufügen und Bestellen der Upstream-Repositorys fertig sind, wählen Sie Speichern.

Wenn Sie ein Gateway-Repository als Upstream-Repository hinzugefügt haben, können Sie einen Paketmanager verwenden, der mit Ihrem lokalen Repository verbunden ist, um Pakete aus dem entsprechenden öffentlichen, externen Paket-Repository abzurufen. Sie müssen Ihre Paketmanager-Konfiguration nicht aktualisieren. Auf diese Weise verbrauchte Pakete werden sowohl im Gateway-Repository als auch in Ihrem lokalen Paket-Repository gespeichert. Weitere Hinweise zum Anfordern von Paketversionen aus einem Upstream-Repository finden Sie unter [Eine Paketversion mit Upstream-Repositorys anfordern](#).

## Unterstützte externe Paket-Repositorys und ihre Gateway-Repositorys

CodeCatalyst unterstützt das Hinzufügen einer Upstream-Verbindung zu den folgenden offiziellen Paketautoritäten mit Gateway-Repositorys.

Repository-Pakettyp	Beschreibung	Name des Gateway-Repositorys
npm	öffentliches Register von npm	npm-public-registry-gateway

## Mit Paketen arbeiten

Ein Paket in CodeCatalyst ist ein Paket aus Software und Metadaten, das zum Auflösen von Abhängigkeiten und zur Installation der Software erforderlich ist. CodeCatalyst unterstützt das npm-Paketformat. Dieser Abschnitt enthält Informationen zum Veröffentlichen, Anzeigen und Löschen von Paketen sowie zum Aktualisieren des Status einer Paketversion.

### Themen

- [Veröffentlichen von Paketen](#)
- [Details zur Paketversion anzeigen](#)
- [Löschen einer Paketversion](#)
- [Den Status einer Paketversion aktualisieren](#)
- [Steuerelemente für den Paketursprung bearbeiten](#)

## Veröffentlichen von Paketen

Sie können Versionen aller unterstützten Pakettypen mithilfe von Paketmanager-Tools in einem CodeCatalyst Paket-Repository veröffentlichen.

Hinweise dazu, wie Sie einen Paketmanager mit einem CodeCatalyst Paket-Repository verbinden, um Pakete zu veröffentlichen, finden Sie unter [Verbindung zu einem Paket-Repository herstellen](#).

### Inhalt

- [Veröffentlichung und Upstream-Repositorys](#)
- [Private Pakete und öffentliche Repositorien](#)
- [Paket-Assets überschreiben](#)

## Veröffentlichung und Upstream-Repositorys

In können Sie keine Paketversionen veröffentlichen CodeCatalyst, die in erreichbaren Upstream-Repositorys oder öffentlichen Repositorys vorhanden sind. Nehmen wir zum Beispiel an, Sie möchten ein npm-Paket in einem Paket-Repository veröffentlichen und es ist über ein Gateway-Repository `myrepo`, das als Upstream-Repository konfiguriert `myrepo` ist, mit `npmjs.com` verbunden. `lodash@1.0` Wenn im Upstream-Repository oder auf `npmjs.com` vorhanden `lodash@1.0` ist, CodeCatalyst lehnt es jeden Versuch ab, es in `npmjs.com` zu veröffentlichen, indem es einen 409-Konfliktfehler ausgibt. `myrepo` Dadurch wird verhindert, dass Sie versehentlich ein Paket mit demselben Namen und derselben Version wie ein Paket in einem Upstream-Repository veröffentlichen, was zu unerwartetem Verhalten führen kann.

Sie können immer noch verschiedene Versionen eines Paketnamens veröffentlichen, die in einem Upstream-Repository existieren. Wenn `lodash@1.0` es beispielsweise in einem Upstream-Repository vorhanden `lodash@1.1` ist, aber nicht, können Sie im Downstream-Repository veröffentlichen `lodash@1.1`.

## Private Pakete und öffentliche Repositorien

CodeCatalyst veröffentlicht keine Pakete, die in Repositorien gespeichert sind, in öffentlichen CodeCatalyst Repositorien wie `npmjs.com`. CodeCatalyst importiert Pakete aus öffentlichen Repositorien in ein CodeCatalyst Projektarchiv, verschiebt Pakete jedoch nicht in die entgegengesetzte Richtung. Pakete, die Sie in CodeCatalyst Repositorys veröffentlichen, bleiben privat und sind nur für das CodeCatalyst Projekt verfügbar, zu dem das Repository gehört.



## Paket-Assets überschreiben

Sie können ein bereits vorhandenes Paket-Asset nicht erneut veröffentlichen, in dem andere Inhalte enthalten sind. Da npm nur ein einzelnes Asset pro Paketversion unterstützt, müssen Sie eine veröffentlichte Paketversion zuerst löschen, um sie zu ändern.

## Details zur Paketversion anzeigen

Sie können die CodeCatalyst Konsole verwenden, um Details zu einer bestimmten Paketversion anzuzeigen.

Um Details zur Paketversion anzuzeigen

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie auf der Seite Paket-Repositoryys das Repository aus, das die Paketversion enthält, deren Details Sie anzeigen möchten.
3. Suchen Sie in der Tabelle Pakete nach der Paketversion. Sie können die Suchleiste verwenden, um Pakete nach Paketnamen zu filtern. Wählen Sie das Paket aus der Liste aus.
4. Wählen Sie auf der Seite mit den Paketdetails die Option Versionen und dann die Version aus, die Sie anzeigen möchten.

## Löschen einer Paketversion

Sie können eine Paketversion auf der Seite mit den Paketversionsdetails in der CodeCatalyst Konsole löschen.

Um eine Paketversion zu löschen

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie auf der Seite Paket-Repositoryys das Repository aus, das die Paketversion enthält, die Sie löschen möchten.
3. Suchen Sie das Paket und wählen Sie es aus der Tabelle aus.
4. Wählen Sie auf der Seite mit den Paketdetails Versionen und wählen Sie die Version aus, die Sie löschen möchten.
5. Wählen Sie auf der Seite mit den Paketversionsdetails die Option Versionsaktionen und dann Löschen aus.
6. Geben Sie Löschen in das Textfeld ein und wählen Sie Löschen.

## Den Status einer Paketversion aktualisieren

Jede Paketversion in CodeCatalyst hat einen Status, der den aktuellen Status und die Verfügbarkeit der Paketversion beschreibt. Sie können den Status der Paketversion in der CodeCatalyst Konsole ändern. Weitere Hinweise zu den möglichen Statuswerten von Paketversionen und deren Bedeutung finden Sie unter [Status der Paketversion](#).

So aktualisieren Sie den Status einer Paketversion

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie auf der Seite Paket-Repositoryys das Repository aus, das die Paketversion enthält, deren Status Sie aktualisieren möchten.
3. Suchen Sie das Paket und wählen Sie es aus der Tabelle aus.
4. Wählen Sie auf der Seite mit den Paketdetails die Option Versionen und dann die Version aus, die Sie anzeigen möchten.
5. Wählen Sie auf der Seite mit den Paketversionsdetails die Option Aktionen und dann Liste aufheben, Archivieren oder Löschen aus. Informationen zum Status der einzelnen Paketversionen finden Sie unter [Status der Paketversion](#)
6. Geben Sie den Bestätigungstext in das Textfeld ein und wählen Sie dann Je nachdem, auf welchen Status Sie aktualisieren, die Liste entfernen, Archivieren oder Löschen aus.

### Status der Paketversion

Die folgenden Werte sind für den Status der Paketversion möglich. Sie können den Status der Paketversion in der Konsole ändern. Weitere Informationen finden Sie unter [Den Status einer Paketversion aktualisieren](#).

- **Veröffentlicht:** Die Paketversion wurde erfolgreich veröffentlicht und kann von einem Paketmanager angefordert werden. Die Paketversion wird in die Paketversionslisten aufgenommen, die an die Paketmanager zurückgegeben werden, z. B. in der Ausgabe von `npm view <package-name> versions`. Alle Ressourcen der Paketversion sind im Repository verfügbar.
- **Nicht gelistet:** Die Paketversions-Assets können aus dem Repository heruntergeladen werden, aber die Paketversion ist nicht in der Liste der Versionen enthalten, die an die Paketmanager zurückgegeben werden. Bei einem npm-Paket enthält die Ausgabe von beispielsweise `npm view <package-name> versions` nicht die Paketversion. Das bedeutet, dass die NPM-Abhängigkeitsauflösungslogik die Paketversion nicht auswählt, da die Version nicht in der Liste

der verfügbaren Versionen erscheint. Wenn die Paketversion „Nicht gelistet“ jedoch bereits in einer `npm package-lock.json` Datei referenziert wird, kann sie trotzdem heruntergeladen und installiert werden, z. B. wenn sie ausgeführt wird. `npm ci`

- **Archiviert:** Die Ressourcen der Paketversion können nicht heruntergeladen werden. Die Paketversion wird nicht in die Liste der Versionen aufgenommen, die an die Paketmanager zurückgegeben werden. Da die Ressourcen nicht verfügbar sind, wird die Nutzung der Paketversion durch Clients blockiert. Wenn der Build Ihrer Anwendung von einer Version abhängt, die auf Archiviert aktualisiert wurde, schlägt der Build fehl, sofern die Paketversion nicht lokal zwischengespeichert wurde. Sie können einen Paketmanager oder ein Build-Tool nicht verwenden, um eine archivierte Paketversion erneut zu veröffentlichen, da sie immer noch im Repository vorhanden ist. Sie können den Status der Paketversion jedoch in der Konsole wieder auf Nicht gelistet oder Veröffentlicht ändern.
- **Verworfen:** Die Paketversion erscheint nicht in den Auflistungen, und die Inhalte können nicht aus dem Repository heruntergeladen werden. Der Hauptunterschied zwischen „Verworfen“ und „Archiviert“ besteht darin, dass bei einem Status von „Verworfen“ die Inhalte der Paketversion dauerhaft von gelöscht werden CodeCatalyst. Aus diesem Grund können Sie eine Paketversion nicht von „Verworfen“ in „Archiviert“, „Nicht gelistet“ oder „Veröffentlicht“ verschieben. Die Paketversion kann nicht verwendet werden, da die Assets gelöscht wurden. Wenn eine Paketversion als entsorgt markiert wurde, wird Ihnen die Aufbewahrung der Paketressourcen nicht in Rechnung gestellt.

Zusätzlich zu den Status in der obigen Liste kann eine Paketversion auch gelöscht werden. Nach dem Löschen befindet sich eine Paketversion nicht mehr im Repository und Sie können diese Paketversion mit einem Paketmanager oder einem Build-Tool nach Belieben erneut veröffentlichen.

## Steuerelemente für den Paketurprung bearbeiten

In Amazon CodeCatalyst können Paketversionen zu einem Paket-Repository hinzugefügt werden, indem sie direkt veröffentlicht, aus einem Upstream-Repository heruntergeladen oder aus einem externen, öffentlichen Repository aufgenommen werden. Wenn Sie zulassen, dass Versionen eines Pakets sowohl durch direkte Veröffentlichung als auch durch Aufnahme aus öffentlichen Repositories hinzugefügt werden, sind Sie anfällig für Angriffe, die Abhängigkeiten ersetzen. Weitere Informationen finden Sie unter [Angriffe durch Substitution von Abhängigkeiten](#). Um sich vor einem Angriff durch die Substitution von Abhängigkeiten zu schützen, konfigurieren Sie die Kontrolle über den Paketurprung für ein Paket in einem Repository, um einzuschränken, wie Versionen dieses Pakets dem Repository hinzugefügt werden können.

Sie sollten in Erwägung ziehen, die Kontrolle über die Herkunft von Paketen so zu konfigurieren, dass neue Versionen verschiedener Pakete sowohl aus internen Quellen wie Direktveröffentlichungen als auch aus externen Quellen wie öffentlichen Repositorien stammen. Standardmäßig werden die Kontrollen für den Paketursprung darauf konfiguriert, wie die erste Version eines Pakets zum Repository hinzugefügt wird.

## Einstellungen für die Kontrolle des Paketursprungs

Mit den Steuerelementen für den Paketursprung können Sie konfigurieren, wie Paketversionen zu einem Repository hinzugefügt werden können. Die folgenden Listen enthalten die verfügbaren Einstellungen und Werte für die Steuerung des Paketursprungs.

### Veröffentlichen

Diese Einstellung konfiguriert, ob Paketversionen mithilfe von Paketmanagern oder ähnlichen Tools direkt im Repository veröffentlicht werden können.

- **ZULASSEN:** Paketversionen können direkt veröffentlicht werden.
- **BLOCK:** Paketversionen können nicht direkt veröffentlicht werden.

### Upstream

Diese Einstellung konfiguriert, ob Paketversionen aus externen, öffentlichen Repositorys aufgenommen oder von Upstream-Repositorys beibehalten werden können, wenn dies von einem Paketmanager angefordert wird.

- **ALLOW:** Jede Paketversion kann aus anderen CodeCatalyst Repositorys beibehalten werden, die als Upstream-Repositorys konfiguriert sind, oder aus einer öffentlichen Quelle mit einer externen Verbindung aufgenommen werden.
- **BLOCKIEREN:** Paketversionen können nicht aus anderen CodeCatalyst Repositorys aufbewahrt werden, die als Upstream-Repositorys konfiguriert sind, oder von einer öffentlichen Quelle mit einer externen Verbindung aufgenommen werden.

## Standardeinstellungen für die Kontrolle des Paketursprungs

Die Standardkontrollen für den Paketursprung für ein Paket basieren darauf, wie die erste Version dieses Pakets dem Paket-Repository hinzugefügt wird.

- Wenn die erste Paketversion direkt von einem Paketmanager veröffentlicht wird, lauten die Einstellungen Publish: ALLOW und Upstream: BLOCK.
- Wenn die erste Paketversion aus einer öffentlichen Quelle aufgenommen wurde, lauten die Einstellungen Publish: BLOCK und Upstream: ALLOW.

## Allgemeine Szenarien zur Paketzugriffskontrolle

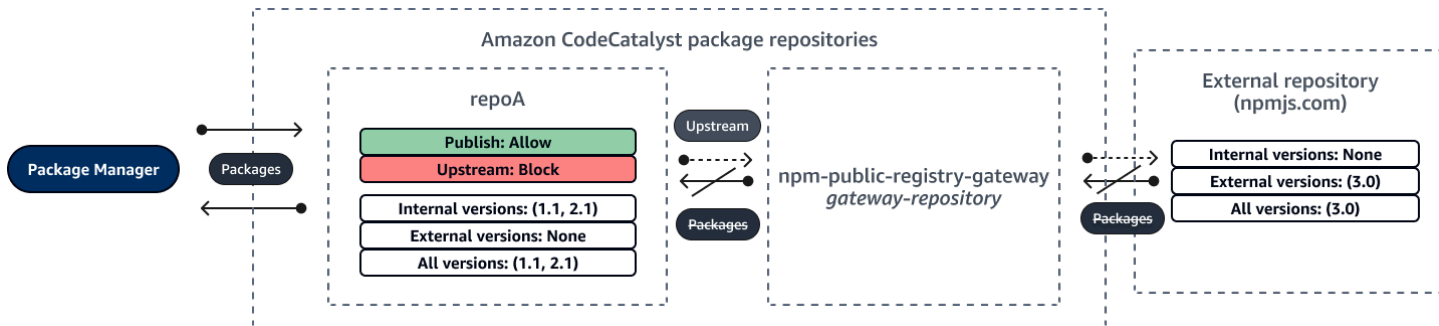
In diesem Abschnitt werden einige gängige Szenarien beschrieben, in denen eine Paketversion zu einem CodeCatalyst Paket-Repository hinzugefügt wird. Die Einstellungen zur Kontrolle des Paketursprungs werden für neue Pakete festgelegt, je nachdem, wie die erste Paketversion hinzugefügt wurde.

In den folgenden Szenarien wird ein internes Paket direkt von einem Paketmanager in Ihrem Repository veröffentlicht, z. B. ein Paket, das Sie verwalten. Ein externes Paket ist ein Paket, das in einem öffentlichen Repository vorhanden ist und über eine externe Verbindung in Ihr Repository aufgenommen werden kann.

Eine externe Paketversion wird für ein vorhandenes internes Paket veröffentlicht

Stellen Sie sich in diesem Szenario ein internes Paket, PackageA, vor. Ihr Team veröffentlicht die erste Paketversion für PackageA in einem Paket-Repository. CodeCatalyst Da dies die erste Paketversion für dieses Paket ist, werden die Einstellungen für die Kontrolle des Paketursprungs automatisch auf Veröffentlichen: Zulassen und Upstream: Blockieren gesetzt. Nachdem das Paket in Ihrem Repository veröffentlicht wurde, wird ein Paket mit demselben Namen in einem öffentlichen Repository veröffentlicht, das mit Ihrem CodeCatalyst Paket-Repository verbunden ist. Dies könnte ein versuchter Angriff zur Substitution von Abhängigkeiten auf das interne Paket sein, oder es könnte ein Zufall sein. Unabhängig davon sind die Kontrollen zur Paketherkunft so konfiguriert, dass sie die Aufnahme der neuen externen Version blockieren, um sich vor einem möglichen Angriff zu schützen.

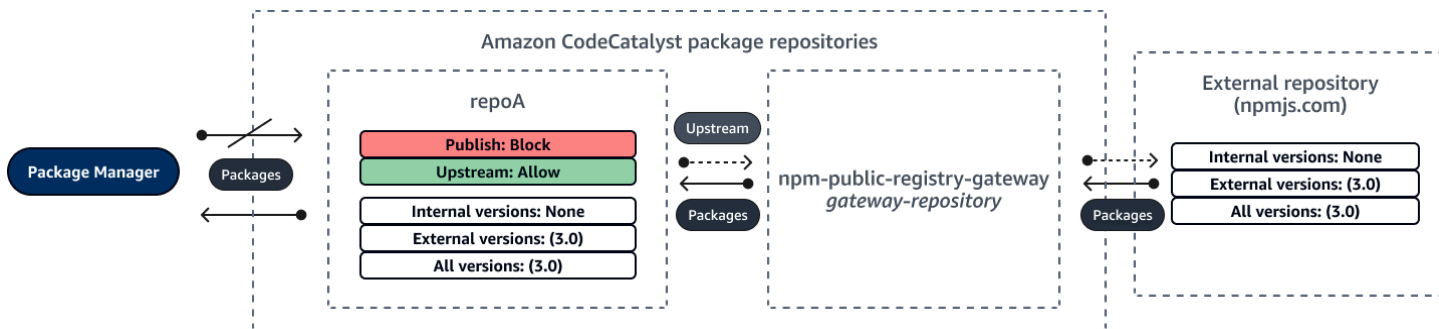
In der folgenden Abbildung ist RepoA Ihr CodeCatalyst Paket-Repository mit einer externen Verbindung zu einem öffentlichen Repository. Ihr Repository enthält die Versionen 1.1 und 2.1 von PackageA, aber Version 3.0 ist im öffentlichen Repository veröffentlicht. Normalerweise würde RepoA Version 3.0 aufnehmen, nachdem das Paket von einem Paketmanager angefordert wurde. Da die Paketaufnahme auf Blockieren gesetzt ist, wird Version 3.0 nicht in Ihr CodeCatalyst Paket-Repository aufgenommen und steht den damit verbundenen Paketmanagern nicht zur Verfügung.



Eine interne Paketversion wird für ein vorhandenes externes Paket veröffentlicht

In diesem Szenario existiert ein Paket, PackageB, extern in einem öffentlichen Repository, das Sie mit Ihrem Repository verbunden haben. Wenn ein mit Ihrem Repository verbundener Paketmanager PackageB anfordert, wird die Paketversion aus dem öffentlichen Repository in Ihr Repository aufgenommen. Da dies die erste Paketversion von PackageB ist, die zu Ihrem Repository hinzugefügt wurde, sind die Einstellungen für den Paketursprung auf **Publish: BLOCK** und **Upstream: ALLOW** konfiguriert. Später versuchen Sie, eine Version mit demselben Paketnamen im Repository zu veröffentlichen. Möglicherweise kennen Sie das öffentliche Paket nicht und versuchen, ein nicht verwandtes Paket mit demselben Namen zu veröffentlichen, oder Sie versuchen möglicherweise, eine gepatchte Version zu veröffentlichen, oder Sie versuchen möglicherweise, genau die Paketversion, die bereits extern existiert, direkt zu veröffentlichen. CodeCatalyst lehnt die Version ab, die Sie zu veröffentlichen versuchen, aber Sie können die Ablehnung explizit außer Kraft setzen und die Version bei Bedarf veröffentlichen.

In der folgenden Abbildung ist RepoA Ihr CodeCatalyst Paket-Repository mit einer externen Verbindung zu einem öffentlichen Repository. Ihr Paket-Repository enthält Version 3.0, die es aus dem öffentlichen Repository aufgenommen hat. Sie möchten Version 1.2 in Ihrem Paket-Repository veröffentlichen. Normalerweise könnten Sie Version 1.2 in RepoA veröffentlichen, aber da die Veröffentlichung auf Blockieren eingestellt ist, kann Version 1.2 nicht veröffentlicht werden.



Veröffentlichen einer gepatchten Paketversion eines vorhandenen externen Pakets

In diesem Szenario existiert ein Paket, PackageB, extern in einem öffentlichen Repository, das Sie mit Ihrem Paket-Repository verbunden haben. Wenn ein mit Ihrem Repository verbundener Paketmanager PackageB anfordert, wird die Paketversion aus dem öffentlichen Repository in Ihr Repository aufgenommen. Da dies die erste Paketversion von PackageB ist, die zu Ihrem Repository hinzugefügt wurde, sind die Einstellungen für den Paketursprung auf Publish: BLOCK und Upstream: ALLOW konfiguriert. Ihr Team beschließt, gepatchte Paketversionen dieses Pakets im Repository zu veröffentlichen. Um Paketversionen direkt veröffentlichen zu können, ändert Ihr Team die Einstellungen zur Kontrolle des Paketursprungs in Publish: ALLOW und Upstream: BLOCK. Versionen dieses Pakets können jetzt direkt in Ihrem Repository veröffentlicht und aus öffentlichen Repositories aufgenommen werden. Nachdem Ihr Team die gepatchten Paketversionen veröffentlicht hat, setzt Ihr Team die Einstellungen für den Paketursprung auf Publish: BLOCK und Upstream: ALLOW zurück.

## Die Einstellungen für den Paketursprung werden bearbeitet

Die Kontrollen zur Paketherkunft werden automatisch konfiguriert, je nachdem, wie die erste Paketversion eines Pakets zum Paket-Repository hinzugefügt wird. Weitere Informationen finden Sie unter [Standardeinstellungen für die Kontrolle des Paketursprungs](#). Gehen Sie wie folgt vor, um Steuerungen für den Paketursprung für ein CodeCatalyst Paket in einem Paket-Repository hinzuzufügen oder zu bearbeiten.

Um Steuerelemente für den Paketursprung hinzuzufügen oder zu bearbeiten

1. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
2. Wählen Sie das Paket-Repository aus, das das Paket enthält, das Sie bearbeiten möchten.
3. Suchen Sie in der Tabelle Pakete nach dem Paket, das Sie bearbeiten möchten, und wählen Sie es aus.
4. Wählen Sie auf der Seite mit der Paketübersicht die Option Ursprungskontrollen bearbeiten aus.
5. Wähle unter Origin-Kontrollen die Kontrollen für den Paketursprung aus, die du für dieses Paket einrichten möchtest. Beide Einstellungen zur Kontrolle des Paketursprungs, Publish und Upstream, müssen gleichzeitig festgelegt werden.
  - Um das direkte Veröffentlichen von Paketversionen zuzulassen, wählen Sie unter Veröffentlichen die Option Zulassen aus. Um die Veröffentlichung von Paketversionen zu blockieren, wählen Sie Blockieren aus.
  - Um die Aufnahme von Paketen aus externen Repositories und das Abrufen von Paketen aus Upstream-Repositories zuzulassen, wählen Sie unter Upstream-Quellen die Option Zulassen



aus. Um die gesamte Aufnahme und das Abrufen von Paketversionen aus externen und Upstream-Repositorys zu blockieren, wählen Sie Blockieren.

6. Wählen Sie Speichern.

## Veröffentlichung und Upstream-Repositorys

In können Sie keine Paketversionen veröffentlichen CodeCatalyst, die in erreichbaren Upstream-Repositorys oder öffentlichen Repositorys vorhanden sind. Nehmen wir zum Beispiel an, Sie möchten ein npm-Paket in einem Repository veröffentlichen und myrepo haben ein Upstream-Repository mit einer externen Verbindung lodash@1.0 zu npmjs.com. myrepo Stellen Sie sich die folgenden Szenarien vor.

1. Die Einstellungen für die Kontrolle des Paketursprungs lodash lauten Publish: ALLOW und Upstream: ALLOW. Wenn im Upstream-Repository oder auf npmjs.com vorhanden lodash@1.0 ist, CodeCatalyst lehnt es jeden Versuch ab, in diesem Verzeichnis zu veröffentlichen, myrepo indem es einen 409-Konfliktfehler ausgibt. Sie könnten immer noch eine andere Version veröffentlichen, z. B. lodash@1.1
2. Die Einstellungen für die Kontrolle des Paketursprungs lodash lauten Publish: ALLOW und Upstream: BLOCK. Sie können jede Version von lodash in Ihrem Repository veröffentlichen, die noch nicht existiert, da Paketversionen nicht erreichbar sind.
3. Die Einstellungen zur Kontrolle des Paketursprungs lodash lauten Publish: BLOCK und Upstream: ALLOW. Sie können keine Paketversionen direkt in Ihrem Repository veröffentlichen.

## Angriffe durch Substitution von Abhängigkeiten

Paketmanager vereinfachen das Verpacken und Teilen von wiederverwendbarem Code. Bei diesen Paketen kann es sich um private Pakete handeln, die von einer Organisation zur Verwendung in ihren Anwendungen entwickelt wurden, oder es kann sich um öffentliche Pakete handeln, in der Regel Open-Source-Pakete, die außerhalb einer Organisation entwickelt und über öffentliche Paket-Repositorys verteilt werden. Bei der Anforderung von Paketen verlassen sich Entwickler auf ihren Paketmanager, um neue Versionen ihrer Abhängigkeiten abzurufen. Angriffe zur Substitution von Abhängigkeiten, auch bekannt als Dependency Confusion Attacks, nutzen die Tatsache aus, dass ein Paketmanager normalerweise keine Möglichkeit hat, legitime Versionen eines Pakets von bösartigen Versionen zu unterscheiden.



Angriffe zur Substitution von Abhängigkeiten gehören zu einer Untergruppe von Angriffen, die als Software-Supply-Chain-Angriffe bezeichnet werden. Ein Angriff auf die Software-Lieferkette ist ein Angriff, bei dem Sicherheitslücken überall in der Software-Lieferkette ausgenutzt werden.

Ein Angriff zur Substitution von Abhängigkeiten kann sich gegen jeden richten, der sowohl intern entwickelte Pakete als auch Pakete verwendet, die aus öffentlichen Repositorien abgerufen wurden. Die Angreifer identifizieren interne Paketnamen und platzieren dann strategisch bösartigen Code mit demselben Namen in öffentlichen Paket-Repositories. In der Regel wird der bösartige Code in einem Paket mit einer hohen Versionsnummer veröffentlicht. Paketmanager rufen den bösartigen Code aus diesen öffentlichen Feeds ab, weil sie glauben, dass es sich bei den bösartigen Paketen um die neuesten Versionen des Pakets handelt. Dies führt zu einer „Verwechslung“ oder „Ersetzung“ zwischen dem gewünschten Paket und dem bösartigen Paket, wodurch der Code kompromittiert wird.

Um Angriffe durch die Substitution von Abhängigkeiten zu verhindern, CodeCatalyst bietet Amazon Kontrollen zur Herkunft von Paketen. Kontrollen zur Paketherkunft sind Einstellungen, die steuern, wie Pakete zu Ihren Repositories hinzugefügt werden können. Die Steuerelemente werden automatisch konfiguriert, wenn die erste Paketversion eines neuen Pakets zu einem CodeCatalyst Repository hinzugefügt wird. Sie können sicherstellen, dass Paketversionen nicht sowohl direkt in Ihrem Repository veröffentlicht als auch aus öffentlichen Quellen aufgenommen werden können, wodurch Sie vor Angriffen durch die Substitution von Abhängigkeiten geschützt sind. Weitere Informationen zu den Kontrollen zur Paketherkunft und zu deren Änderung finden Sie unter [Steuerelemente für den Paketursprung bearbeiten](#)

## Verwenden von npm

In diesen Themen wird beschrieben, wie Sie den Paketmanager Node.js mit CodeCatalyst verwenden können.

### Note

CodeCatalyst unterstützt `node v4.9.1` und später `npm v5.0.0` und später.

### Themen

- [Konfiguration und Verwendung von npm](#)
- [Handhabung von NPM-Tags](#)

## Konfiguration und Verwendung von npm

Um npm mit zu verwenden CodeCatalyst, müssen Sie eine Verbindung npm zu Ihrem Paket-Repository herstellen und ein Personal Access Token (PAT) für die Authentifizierung bereitstellen. Anweisungen zum Herstellen einer Verbindung npm zu Ihrem Paket-Repository finden Sie in der CodeCatalyst Konsole.

### Inhalt

- [Konfiguration von npm mit CodeCatalyst](#)
- [NPM-Pakete aus einem Paket-Repository installieren CodeCatalyst](#)
- [Installation von NPM-Paketen von npmjs bis CodeCatalyst](#)
- [Veröffentlichen Sie npm-Pakete in Ihrem CodeCatalyst Paket-Repository](#)
- [Unterstützung für npm-Befehle](#)
  - [Unterstützte Befehle, die mit einem Paket-Repository interagieren](#)
  - [Unterstützte clientseitige Befehle](#)
  - [Befehle werden nicht unterstützt](#)


### Konfiguration von npm mit CodeCatalyst

In den folgenden Anweisungen wird erklärt, wie Sie sich authentifizieren und eine Verbindung npm zu Ihrem CodeCatalyst Paket-Repository herstellen. Weitere Informationen zu npm finden Sie in der [offiziellen npm-Dokumentation](#).

Um eine Verbindung **npm** zu Ihrem Paket-Repository CodeCatalyst herzustellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt.
3. Wählen Sie im Navigationsbereich Packages (Pakete) aus.
4. Wählen Sie Ihr Paket-Repository aus der Liste aus.
5. Wählen Sie Mit Repository verbinden.
6. Wählen Sie in den Konfigurationsdetails im Package Manager Client die Option npm client aus.
7. Wählen Sie Ihr Betriebssystem aus, um die entsprechenden Konfigurationsschritte anzuzeigen.
8. Für die Authentifizierung von npm ist ein Personal Access Token (PAT) erforderlich. CodeCatalyst Wenn Sie bereits ein Token haben, können Sie es verwenden. Wenn nicht, können Sie eines mit den folgenden Schritten erstellen.


- a. (Optional): Aktualisieren Sie den PAT-Namen und das Ablaufdatum.
- b. Wählen Sie „Token erstellen“.
- c. Kopieren Sie Ihre PAT und speichern Sie sie an einem sicheren Ort.

 Warning

Sie können Ihr PAT nicht mehr sehen oder kopieren, nachdem Sie das Dialogfeld geschlossen haben. Anmeldeinformationen sollten kurzlebig sein, um die Zeit zu minimieren, für die ein Angreifer die Anmeldeinformationen verwenden kann, nachdem sie missbraucht wurden.

9. Führen Sie die folgenden Befehle im Stammverzeichnis Ihres Projekts aus, um npm mit Ihrem Paket-Repository zu konfigurieren. Die Befehle bewirken Folgendes:
  - Erstellen Sie eine `.npmrc` Datei auf Projektebene, falls Ihr Projekt noch keine hat.
  - Fügen Sie die Endpunktinformationen des Paket-Repositorys zu Ihrer Datei auf `.npmrc` Projektebene hinzu.
  - Fügen Sie Ihre Anmeldeinformationen (PAT) zu Ihrer Datei auf `.npmrc` Benutzerebene hinzu.

Ersetzen Sie die folgenden Werte.

 Note

Wenn Sie Anweisungen aus der Konsole kopieren, werden die Werte in den folgenden Befehlen für Sie aktualisiert und müssen nicht geändert werden.

- Ersetzen Sie den *Benutzernamen* durch Ihren CodeCatalyst Benutzernamen.
- Ersetzen Sie *PAT* durch Ihr CodeCatalyst PAT.
- Ersetzen Sie *space\_name* durch Ihren CodeCatalyst Space-Namen.
- Ersetzen Sie *proj\_name* durch *Ihren Projektnamen*. CodeCatalyst
- Ersetzen Sie *repo\_name* durch *den Namen Ihres Paket-Repositorys*. CodeCatalyst

```
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
```

Für npm 6 oder niedriger: Damit npm das Auth-Token auch bei GET Anfragen immer weitergibt CodeCatalyst, setzen Sie die Konfigurationsvariable `always-auth` wie folgt. `npm config set`

```
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:always-auth=true --location project
```

## NPM-Pakete aus einem Paket-Repository installieren CodeCatalyst

Nachdem Sie npm mit Ihrem Repository verbunden haben, indem Sie die Schritte unter ausgeführt haben [Konfiguration von npm mit CodeCatalyst](#), können Sie npm Befehle in Ihrem Repository ausführen.

Sie können ein npm-Paket, das sich in Ihrem CodeCatalyst Paket-Repository oder einem seiner Upstream-Repositorys befindet, mit dem Befehl installieren. `npm install`

```
npm install lodash
```

## Installation von NPM-Paketen von npmjs bis CodeCatalyst

Sie können npm-Pakete von [npmjs.com](#) über ein CodeCatalyst Repository installieren, indem Sie das Repository mit einer Upstream-Verbindung zum Gateway-Repository konfigurieren, das mit npmjs.com verbunden ist. npm-public-registry-gateway Pakete, die über npmjs installiert wurden, werden im Gateway-Repository und im Paket-Repository am weitesten unten aufgenommen und gespeichert.

Um Pakete von npmjs zu installieren

1. Falls Sie dies noch nicht getan haben, konfigurieren Sie es npm mit Ihrem CodeCatalyst Paket-Repository, indem Sie die Schritte unter befolgen. [Konfiguration von npm mit CodeCatalyst](#)
2. Vergewissern Sie sich, dass Ihr Repository das Gateway-Repository, npm-public-registry-gateway, als Upstream-Verbindung hinzugefügt hat. Sie können überprüfen, welche Upstream-

Quellen hinzugefügt oder npm-public-registry-gateways Upstream-Quelle hinzugefügt wurden, indem Sie den Anweisungen unter folgen [Hinzufügen eines Upstream-Repositorys](#) und das npm-public-registry-gatewayRepository auswählen.

3. Installieren Sie Pakete mit dem `npm install` Befehl.

```
npm install package_name
```

Weitere Hinweise zum Anfordern von Paketen aus Upstream-Repositorys finden Sie unter [Eine Paketversion mit Upstream-Repositorys anfordern](#).

## Veröffentlichen Sie npm-Pakete in Ihrem CodeCatalyst Paket-Repository

Nachdem Sie den Vorgang abgeschlossen haben [Konfiguration von npm mit CodeCatalyst](#), können Sie npm Befehle ausführen.

Mit dem `npm publish` Befehl können Sie ein npm-Paket in einem CodeCatalyst Paket-Repository veröffentlichen.

```
npm publish
```

Informationen zum Erstellen von npm-Paketen finden Sie unter [Creating Node.js Modules](#) in npm Docs.

## Unterstützung für npm-Befehle

In den folgenden Abschnitten werden die npm Befehle zusammengefasst, die von CodeCatalyst Paket-Repositorys unterstützt werden, sowie spezifische Befehle aufgeführt, die nicht unterstützt werden.

### Themen

- [Unterstützte Befehle, die mit einem Paket-Repository interagieren](#)
- [Unterstützte clientseitige Befehle](#)
- [Befehle werden nicht unterstützt](#)

### Unterstützte Befehle, die mit einem Paket-Repository interagieren

In diesem Abschnitt sind npm Befehle aufgeführt, bei denen der npm Client eine oder mehrere Anfragen an die Registry stellt, für die er konfiguriert ist (z. B. `npm config set registry`). Es

wurde überprüft, ob diese Befehle korrekt funktionieren, wenn sie in einem CodeCatalyst Paket-Repository aufgerufen werden.

Befehl	Beschreibung
<a href="#">Bugs</a>	Errät die Adresse der Bugtracker-URL eines Pakets und versucht dann, es zu öffnen.
<a href="#">ci</a>	Installiert ein Projekt mit einem Neuanfang.
<a href="#">verwerfen</a>	Weist eine Version eines Pakets ab.
<a href="#">dist-tag</a>	Ändert die Tags für die Paketverteilung.
<a href="#">Dokumente</a>	Errät den Speicherort der Dokumentations-URL eines Pakets und versucht dann, es mithilfe des <code>--browser</code> Konfigurationsparameters zu öffnen.
<a href="#">Arzt</a>	Führt eine Reihe von Prüfungen durch, um zu überprüfen, ob Ihre npm-Installation Ihre JavaScript Pakete verwalten kann.
<a href="#">install</a>	Installiert ein Paket.
<a href="#">install-ci-test</a>	Installiert ein völlig neues Projekt und führt Tests durch. Alias: <code>npm ci</code> . Dieser Befehl führt einen <code>ausnpm ci</code> , unmittelbar gefolgt von einem <code>npm test</code> .
<a href="#">installieren/testen</a>	Installiert das Paket und führt Tests aus. Führt einen <code>ausnpm install</code> , unmittelbar gefolgt von einem <code>npm test</code> .
<a href="#">veraltet</a>	Überprüft die konfigurierte Registrierung, um festzustellen, ob installierte Pakete veraltet sind.

Befehl	Beschreibung
<a href="#">pingen</a>	Pingt die konfigurierte oder angegebene npm-Registrierung an und überprüft die Authentifizierung.
<a href="#">veröffentlichen</a>	Veröffentlicht eine Paketversion in der Registrierung.
<a href="#">update</a>	Errät den Speicherort der Repository-URL eines Pakets und versucht dann, es mithilfe des <code>--browser</code> Konfigurationsparameters zu öffnen.
<a href="#">anzeigen</a>	Zeigt Paket-Metadaten an. Kann auch zum Drucken von Metadateneigenschaften verwendet werden.

### Unterstützte clientseitige Befehle

Diese Befehle erfordern keine direkte Interaktion mit einem Paket-Repository und benötigen daher CodeCatalyst nichts, um sie zu unterstützen.

Befehl	Beschreibung
<a href="#">bin (veraltet)</a>	Zeigt das bin NPM-Verzeichnis an.
<a href="#">bauen</a>	Baut ein Paket.
<a href="#">Cache</a>	Manipuliert den Paket-Cache.
<a href="#">Abschluss</a>	Aktiviert die Tabulatorvervollständigung in allen npm-Befehlen.
<a href="#">Konfiguration</a>	Aktualisiert den Inhalt der Benutzer- und <code>npmrc</code> Globaldateien.
<a href="#">deduplizieren</a>	Durchsucht den lokalen Paketbaum und versucht, die Struktur zu vereinfachen, indem

Befehl	Beschreibung
	Abhängigkeiten im Baum weiter nach oben verschoben werden, wo sie effektiver von mehreren abhängigen Paketen gemeinsam genutzt werden können.
<a href="#">bearbeiten</a>	Bearbeitet ein installiertes Paket. Wählt eine Abhängigkeit im aktuellen Arbeitsverzeichnis aus und öffnet das Paketverzeichnis im Standardeditor.
<a href="#">erkunden</a>	Durchsucht ein installiertes Paket. Erzeugt eine Subshell im Verzeichnis des angegebenen installierten Pakets. Wenn ein Befehl angegeben ist, wird er in der Subshell ausgeführt, die dann sofort heruntergefahren wird.
<a href="#">help</a>	Ruft Hilfe zu npm ab.
<a href="#">Hilfesuche</a>	Durchsucht die NPM-Hilfedokumentation.
<a href="#">init</a>	Erzeugt eine <code>package.json</code> Datei.
<a href="#">Link</a>	Verweist symbolisch auf ein Paketverzeichnis.
<a href="#">ls</a>	Listet die installierten Pakete auf.
<a href="#">packen</a>	Erzeugt einen Tarball aus einem Paket.
<a href="#">prefix</a>	Zeigt ein Präfix an. Dies ist das nächstgelegene übergeordnete Verzeichnis, das eine <code>package.json</code> Datei enthält, sofern es nicht ebenfalls angegeben <code>-g</code> ist.
<a href="#">pflaumen</a>	Entfernt Pakete, die nicht in der Abhängigkeitsliste des übergeordneten Pakets aufgeführt sind.



Befehl	Beschreibung
<a href="#">neu aufbauen</a>	Führt den <code>npm build</code> Befehl für die entsprechenden Ordner aus.
<a href="#">neu starten</a>	Führt die Stopp-, Restart- und Start-Skripten eines Pakets sowie die zugehörigen Vor- und Nachskripte aus.
<a href="#">Root</a>	Druckt das aktuelle <code>node_modules</code> Verzeichnis auf die Standardausgabe aus.
<a href="#">Run-Skript</a>	Führt beliebige Paketskripten aus.
<a href="#">shrinkwrap</a>	Sperrt abhängige Versionen für die Veröffentlichung.
<a href="#">deinstallieren</a>	Deinstalliert ein Paket.

Befehle werden nicht unterstützt

Diese npm Befehle werden von CodeCatalyst Paket-Repositorys nicht unterstützt.

Befehl	Beschreibung	Hinweise
<a href="#">Zugriff</a>	Legt die Zugriffsebene für veröffentlichte Pakete fest.	CodeCatalyst verwendet ein Berechtigungsmodell, das sich vom öffentlichen npmjs-Repository unterscheidet.
<a href="#">Benutzer hinzufügen</a>	Fügt ein Registrierungsbenutzerkonto hinzu	CodeCatalyst verwendet ein Benutzermodell, das sich vom öffentlichen npmjs-Repository unterscheidet.
<a href="#">Prüfung</a>	Führt ein Sicherheitsaudit durch.	CodeCatalyst verkauft derzeit keine Daten zu Sicherheitslücken.

Befehl	Beschreibung	Hinweise
<a href="#">Haken</a>	Verwaltet NPM-Hooks, einschließlich Hinzufügen, Entfernen, Auflisten und Aktualisieren.	CodeCatalyst unterstützt derzeit keinen Mechanismus zur Benachrichtigung über Änderungen.
<a href="#">login</a>	Authentifiziert einen Benutzer. Dies ist ein Alias für <code>npm adduser</code> .	CodeCatalyst verwendet ein Authentifizierungsmodell, das sich vom öffentlichen npmjs-Repository unterscheidet. Weitere Informationen finden Sie unter <a href="#">Konfiguration von npm mit CodeCatalyst</a> .
<a href="#">logout</a>	Meldet sich von der Registrierung ab.	CodeCatalyst verwendet ein Authentifizierungsmodell, das sich vom öffentlichen npmjs-Repository unterscheidet. Es gibt keine Möglichkeit, sich von einem CodeCatalyst Repository abzumelden, aber Authentifizierungstoken laufen nach ihrer konfigurierbaren Ablaufzeit ab. Die Standarddauer des Tokens beträgt 12 Stunden.
<a href="#">Besitzer</a>	Verwaltet Paketbesitzer.	CodeCatalyst verwendet ein Berechtigungsmodell, das sich vom öffentlichen npmjs-Repository unterscheidet.
<a href="#">profile</a>	Ändert die Einstellungen in Ihrem Registrierungsprofil.	CodeCatalyst verwendet ein Benutzermodell, das sich vom öffentlichen npmjs-Repository unterscheidet.

Befehl	Beschreibung	Hinweise
<a href="#">search</a>	Durchsucht die Registrierung nach Paketen, die den Suchbegriffen entsprechen.	CodeCatalyst unterstützt den <code>search</code> Befehl nicht.
<a href="#">Stern</a>	Markiert deine Lieblingspakete.	CodeCatalyst unterstützt derzeit keinen Favoriten-Mechanismus.
<a href="#">Sterne</a>	Zeigt Pakete an, die als Favoriten markiert sind.	CodeCatalyst unterstützt derzeit keinen Favoriten-Mechanismus.
<a href="#">Team</a>	Verwaltet Teams und Teammitgliedschaften.	CodeCatalyst verwendet ein Benutzer- und Gruppenmitgliedschaftsmodell, das sich vom öffentlichen npmjs-Repository unterscheidet.
<a href="#">Token</a>	Verwaltet Ihre Authentifizierungstoken.	CodeCatalyst verwendet ein anderes Modell zum Abrufen von Authentifizierungstoken. Weitere Informationen finden Sie unter <a href="#">Konfiguration von npm mit CodeCatalyst</a> .
<a href="#">Veröffentlichung rückgängig machen</a>	Entfernt ein Paket aus der Registrierung.	CodeCatalyst unterstützt nicht das Entfernen einer Paketversion aus einem Repository mithilfe des npm-Clients. Sie können ein Paket in der Konsole löschen.
<a href="#">Whoami</a>	Zeigt den NPM-Benutzernamen an.	CodeCatalyst verwendet ein Benutzermodell, das sich vom öffentlichen npmjs-Repository unterscheidet.

## Handhabung von NPM-Tags

npm-Registries unterstützen Tags, bei denen es sich um Zeichenkettenalias für Paketversionen handelt. Sie können Tags verwenden, um einen Alias bereitzustellen, anstatt Versionsnummern zu verwenden. Sie haben beispielsweise ein Projekt mit mehreren Entwicklungsströmen und verwenden für jeden Stream ein anderes Tag (z. B., `stable`, `beta`, `dev`, `canary`). Weitere Informationen finden Sie unter [dist-tag in npm Docs](#).

Standardmäßig verwendet npm das `latest` Tag, um die aktuelle Version eines Pakets zu identifizieren. `npm install pkg` (ohne `@version` oder `@tag` Spezifizierer) installiert das neueste Tag. In der Regel verwenden Projekte nur das neueste Tag für stabile Release-Versionen. Andere Tags werden für instabile Versionen oder Vorabversionen verwendet.

### Bearbeiten von Tags mit dem NPM-Client

Die drei `npm dist-tag` Befehle (`add`, `rm`, `undls`) funktionieren in CodeCatalyst Paket-Repositorys genauso wie in der [NPM-Standardregistrierung](#).

### npm-Tags und Upstream-Repositorys

Wenn npm Anfragen die Tags für ein Paket enthält und Versionen dieses Pakets auch in einem Upstream-Repository vorhanden sind, werden die Tags CodeCatalyst zusammengeführt, bevor sie an den Client zurückgegeben werden. Ein Projektarchiv mit dem Namen R hat beispielsweise ein Upstream-Repository mit dem Namen U. Die folgende Tabelle zeigt die Tags für ein Paket mit dem Namen `web-helper`, das in beiden Repositorys vorhanden ist.

Repository	Package name	Paket-Tags
R	<code>web-helper</code>	aktuell (Alias für Version 1.0.0)
U	<code>web-helper</code>	alpha (Alias für Version 1.0.1)

In diesem Fall empfängt der npm-Client, wenn er die Tags für das `web-helper` Paket aus dem Repository abrufen, sowohl die neuesten als auch die Alpha-Tags. Die Versionen, auf die die Tags verweisen, werden sich nicht ändern.

Wenn dasselbe Tag für dasselbe Paket sowohl im Upstream-Repository als auch im lokalen Repository vorhanden ist, CodeCatalyst wird das Tag verwendet, das zuletzt aktualisiert wurde.

Nehmen wir zum Beispiel an, dass die Tags auf Webhelper so geändert wurden, dass sie wie folgt aussehen.

Repository	Package name	Paket-Tags	Letzte Aktualisierung
R	web-helper	aktuell (Alias für Version 1.0.0)	1. Januar 2023
U	web-helper	aktuell (Alias für Version 1.0.1)	01. Juni 2023

Wenn in diesem Fall der npm-Client die Tags für das Paket web-helper aus dem Repository abrufft, verwendet das neueste Tag einen Alias für die Version 1.0.1, da es zuletzt aktualisiert wurde. Dies macht es einfach, neue Paketversionen in einem Upstream-Repository, die noch nicht in einem lokalen Repository vorhanden sind, durch Ausführen zu nutzen. `npm update`

## Kontingente für Pakete

In der folgenden Tabelle werden Kontingente und Beschränkungen für Pakete bei Amazon beschrieben CodeCatalyst. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

Ressource	Standardkontingent
Paket-Repositorys	Maximal 1000 pro Speicherplatz.
Direkte Upstream-Repositorys	Maximal 10 pro Paket-Repository.
Upstream-Paket-Repositorys wurden durchsucht	Pro angeforderter Paketversion wurden maximal 25 Upstream-Repositorys durchsucht.
Größe der Paket-Asset-Datei	Maximal 5 GB pro Paket-Asset.
Paket-Assets	Maximal 150 pro Paketversion.

# Erstellen, Testen und Bereitstellen mit Workflows in CodeCatalyst

Nachdem Sie Ihren Anwendungscode in einer [CodeCatalystEntwicklungsumgebung](#) geschrieben und in Ihr [CodeCatalyst Quell-Repository](#) übertragen haben, können Sie ihn bereitstellen. Dies erfolgt automatisch über einen Workflow.

Ein Workflow ist ein automatisiertes Verfahren, das beschreibt, wie Sie Ihren Code als Teil eines CI/CD-Systems (Continuous Integration and Continuous Delivery) erstellen, testen und bereitstellen. Ein Workflow definiert eine Reihe von Schritten oder Aktionen, die während einer Workflow-Ausführung ausgeführt werden sollen. Ein Workflow definiert auch die Ereignisse oder Auslöser, die den Start des Workflows auslösen. Um einen Workflow einzurichten, erstellen Sie mit dem [visuellen Editor oder dem YAML-Editor](#) der CodeCatalyst Konsole eine Workflow-Definitionsdatei.

## Tip

Um einen schnellen Überblick darüber zu erhalten, wie Sie Workflows in einem Projekt verwenden könnten, [erstellen Sie ein Projekt mit einem Blueprint](#). Jeder Blueprint stellt einen funktionierenden Workflow bereit, den Sie überprüfen, ausführen und mit dem Sie experimentieren können.

## Informationen zur Workflow-Definitionsdatei

Eine Workflow-Definitionsdatei ist eine YAML-Datei, die Ihren Workflow beschreibt. Die Datei wird in einem `~/ .codecatalyst/workflows/` Ordner im Stammverzeichnis Ihres [Quell-Repositorys](#) gespeichert. Die Datei kann die Erweiterung `.yml` oder `.yaml` haben.

Im Folgenden finden Sie ein Beispiel für eine einfache Workflow-Definitionsdatei. Wir erklären jede Zeile dieses Beispiels in der folgenden Tabelle.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

**Actions:****Build:**

Identifizier: aws/build@v1

**Inputs:****Sources:**

- WorkflowSource

**Configuration:****Steps:**

- Run: docker build -t MyApp:latest .

Linien	Beschreibung
<pre>Name: MyWorkflow</pre>	<p>Gibt den Namen des Workflows an. Weitere Informationen zu der Name Eigenschaft finden Sie unter <a href="#">Immobilien auf höchstem Niveau</a>.</p>
<pre>SchemaVersion: 1.0</pre>	<p>Gibt die Version des Workflow-Schemas an. Weitere Informationen zu der SchemaVersion Eigenschaft finden Sie unter <a href="#">Immobilien auf höchstem Niveau</a>.</p>
<pre>RunMode: QUEUED</pre>	<p>Gibt an, wie CodeCatalyst mit mehreren Durchläufen umgegangen wird. Weitere Hinweise zum Ausführungsmodus finden Sie unter <a href="#">Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen</a>.</p>
<pre>Triggers:</pre>	<p>Gibt die Logik an, nach der eine Workflow-Ausführung gestartet wird. Weitere Informationen zu Auslösern finden Sie unter <a href="#">Arbeiten mit Auslösern</a>.</p>
<pre>- Type: PUSH   Branches:     - main</pre>	<p>Gibt an, dass der Workflow immer dann gestartet werden muss, wenn Sie Code in den main Zweig des Standard-Quell-Repositorys übertragen. Weitere Informationen zur Workflow-Quelle finden Sie unter <a href="#">Mit Quellen arbeiten</a>.</p>

Linien	Beschreibung
<pre>Actions:</pre>	<p>Definiert die Aufgaben, die während einer Workflow-Ausführung ausgeführt werden müssen. In diesem Beispiel definiert der <code>Actions</code> Abschnitt eine einzelne Aktion namens <code>Build</code>. Weitere Informationen zu Aktionen finden Sie unter <a href="#">Verwenden von Aktionen</a>.</p>
<pre>Build:</pre>	<p>Definiert die Eigenschaften für die <code>Build</code> Aktion. Weitere Hinweise zur <code>Build</code>-Aktion finden Sie unter <a href="#">Erstellen mithilfe von Workflows in CodeCatalyst</a>.</p>
<pre>Identifizier: aws/build@v1</pre>	<p>Gibt den eindeutigen, hartcodierten Bezeichner für die <code>Build</code>-Aktion an.</p>
<pre>Inputs:   Sources:     - WorkflowSource</pre>	<p>Gibt an, dass die <code>Build</code>-Aktion im <code>WorkflowSource</code> Quell-Repository nach den Dateien suchen soll, die sie zum Abschließen der Verarbeitung benötigt. Weitere Informationen finden Sie unter <a href="#">Mit Quellen arbeiten</a>.</p>
<pre>Configuration:</pre>	<p>Enthält die Konfigurationseigenschaften, die für die <code>Build</code>-Aktion spezifisch sind.</p>
<pre>Steps:   - Run: docker build -t MyApp:latest .</pre>	<p>Weist die <code>Build</code>-Aktion an, ein Docker-Image namens <code>MyApp</code> zu erstellen und es mit <code>latest</code> zu kennzeichnen.</p>

Eine vollständige Liste aller in der Workflow-Definitionsdatei verfügbaren Eigenschaften finden Sie unter [Referenz zur Workflow-Definition](#).



# Verwenden der visuellen Editoren und der YAML-Editoren der CodeCatalyst Konsole

Um die Workflow-Definitionsdatei zu erstellen und zu bearbeiten, können Sie Ihren bevorzugten Editor verwenden. Wir empfehlen jedoch, den visuellen Editor oder den YAML-Editor der CodeCatalyst Konsole zu verwenden. Diese Editoren bieten eine hilfreiche Dateiüberprüfung, um sicherzustellen, dass YAML-Eigenschaftsnamen, Werte, Verschachtelungen, Abstände, Groß-/Kleinschreibung usw. korrekt sind.

Die folgende Abbildung zeigt einen Arbeitsablauf im Visual Editor. Der visuelle Editor bietet Ihnen eine vollständige Benutzeroberfläche, über die Sie Ihre Workflow-Definitionsdatei erstellen und konfigurieren können. Der visuelle Editor umfasst ein Workflow-Diagramm (1), das die Hauptkomponenten des Workflows zeigt, und einen Konfigurationsbereich (2).

The screenshot displays the Amazon CodeCatalyst Visual Editor interface. At the top, there's a header with 'ExampleWorkflow', 'ExampleRepository', 'main', and 'Default' buttons, along with 'Cancel', 'Validate', and 'Commit' buttons. Below the header, there are tabs for 'Visual' and 'YAML'. The main area is divided into two sections: 'Workflow diagram' (1) and 'Configuration area' (2). The 'Workflow diagram' shows a vertical flowchart with four steps: 'Source' (ExampleRepository main), 'Triggers' (Push), 'Test' (aws/managed-test@v1), 'BuildBackend' (aws/build@v1, Environment: ExampleEnvironment, Production), and 'DeployCloudFormationStack' (aws/cfn-deploy@v1, Environment: ExampleEnvironment, Production). The 'Configuration area' is for the 'BuildBackend' action and includes sections for 'Inputs', 'Configuration', and 'Outputs'. The 'Configuration' section includes fields for 'Action name' (BuildBackend), 'Compute type' (EC2), 'Compute fleet - optional' (Choose compute), 'Environment/account/role - optional' (ExampleEnvironment), and 'AWS account connection' (111122223333).

Alternativ können Sie den YAML-Editor verwenden, der im nächsten Bild gezeigt wird. Verwenden Sie den YAML-Editor, um große Codeblöcke einzufügen (z. B. aus einem Tutorial) oder um erweiterte Eigenschaften hinzuzufügen, die nicht im visuellen Editor angeboten werden.

```

28 BuildBackend:
29   Identifier: aws/build@v1
30   DependsOn:
31     - Test
32   Environment:
33   Connections:
34     - Role: CodeCatalystPreviewDevelopmentAdministrator-123abc
35       Name: "111122223333"
36   Name: ExampleEnvironment
37   Inputs:
38   Sources:
39     - WorkflowSource
40   Configuration:
41   Steps:
42     - Run: ./setup-sam.sh
43     - Run: sam package --template-file sam-template.yml --s3-bucket
44           codecatalyst-cfn-s3-bucket --output-template-file
45           sam-template-packaged.yml --region us-west-2
46   Outputs:
47   Artifacts:
48     - Name: buildArtifact
49   Files:
50     - "*"/*"
51   Compute:
52     Type: EC2
53 DeployCloudFormationStack:
54   Identifier: aws/cfn-deploy@v1
55   Configuration:
56     capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
57     role-arn: arn:aws:iam::111122223333:role/codecatalyst-stack-role
58     template: ./sam-template-packaged.yml
59     region: us-west-2
60     name: codecatalyst-cfn-stack
61   Environment:
62   Connections:

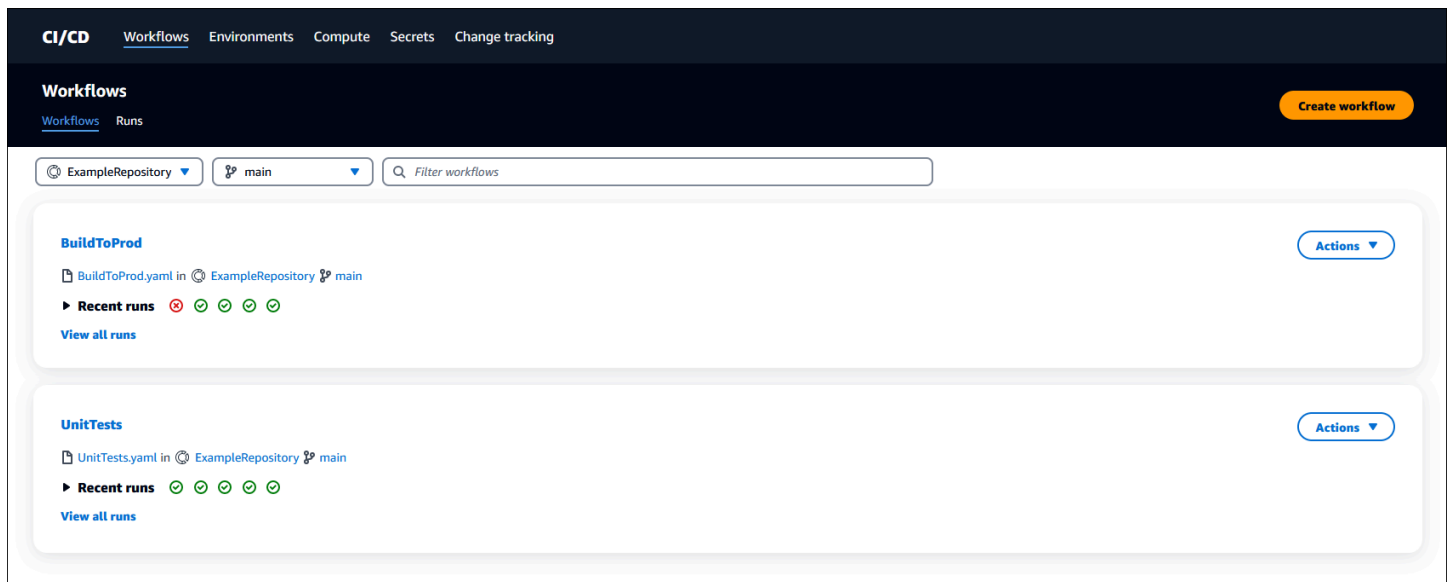
```

Sie können vom visuellen Editor zum YAML-Editor wechseln, um zu sehen, welche Auswirkungen Ihre Konfigurationen auf den zugrunde liegenden YAML-Code haben.

## Workflows entdecken

Sie können Ihren Workflow zusammen mit anderen Workflows, die Sie im selben Projekt eingerichtet haben, auf der Workflow-Übersichtsseite einsehen.

Die folgende Abbildung zeigt die Workflow-Übersichtsseite. Sie enthält zwei Workflows: BuildToProd und UnitTests. Sie können sehen, dass beide einige Male ausgeführt wurden. Sie können Letzte Läufe auswählen, um schnell den Ausführungsverlauf anzuzeigen, oder den Namen des Workflows wählen, um den YAML-Code des Workflows und andere detaillierte Informationen anzuzeigen.



## Details zur Workflow-Ausführung anzeigen

Sie können die Details einer Workflow-Ausführung anzeigen, indem Sie die Ausführung auf der Workflow-Übersichtsseite auswählen.

Die folgende Abbildung zeigt die Details einer Workflow-Ausführung mit der Bezeichnung run-CC11D, die bei einem Commit to Source automatisch gestartet wurde. Das Workflow-Diagramm zeigt, dass eine Aktion fehlgeschlagen ist (1). Sie können zu den Protokollen (2) navigieren, um die detaillierten Protokollmeldungen anzuzeigen und Probleme zu beheben. Weitere Informationen zu Workflow-Ausführungen finden Sie unter [Mit Läufen arbeiten](#).

The screenshot displays the Amazon CodeCatalyst interface for a workflow named 'Run-cc11d'. The workflow is in a 'Failed' state. The 'BuildBackend' action is highlighted as the failed action. A detailed log for 'BuildBackend' shows the following error message:

```

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 /codecatalyst/output/src3862/src/git-codecommit.us

```

## Nächste Schritte

Weitere Informationen zu Workflow-Konzepten finden Sie unter [Workflow-Konzepte](#).

Informationen zum Erstellen Ihres ersten Workflows finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#).

## Workflow-Konzepte

Im Folgenden finden Sie einige Konzepte und Begriffe, die Sie kennen sollten, wenn Sie Ihren Code mit Workflows erstellen, testen oder bereitstellen CodeCatalyst.

## Workflows

Ein Workflow ist ein automatisiertes Verfahren, das beschreibt, wie Sie Ihren Code als Teil eines CI/CD-Systems (Continuous Integration and Continuous Delivery) erstellen, testen und bereitstellen. Ein Workflow definiert eine Reihe von Schritten oder Aktionen, die während einer Workflow-Ausführung ausgeführt werden sollen. Ein Workflow definiert auch die Ereignisse oder Auslöser, die den Start des Workflows auslösen. Um einen Workflow einzurichten, erstellen Sie mit dem [visuellen Editor oder dem YAML-Editor](#) der CodeCatalyst Konsole eine Workflow-Definitionsdatei.

**Tip**

Um einen kurzen Überblick darüber zu erhalten, wie Sie Workflows in einem Projekt verwenden könnten, [erstellen Sie ein Projekt mit einem Blueprint](#). Jeder Blueprint stellt einen funktionierenden Workflow bereit, den Sie überprüfen, ausführen und mit dem Sie experimentieren können.

Weitere Informationen zu Workflows finden Sie unter [Mit Workflows arbeiten](#).

## Workflow-Definitionsdateien

Eine Workflow-Definitionsdatei ist eine YAML-Datei, die Ihren Workflow beschreibt. Die Datei wird in einem `~/.codecatalyst/workflows/` Ordner im Stammverzeichnis Ihres [Quell-Repositorys](#) gespeichert. Die Datei kann die Erweiterung `.yml` oder `.yaml` haben.

Weitere Informationen zur Workflow-Definitionsdatei finden Sie unter [Referenz zur Workflow-Definition](#)

## Aktionen

Eine Aktion ist der Hauptbaustein eines Workflows und definiert eine logische Arbeitseinheit, die während einer Workflow-Ausführung ausgeführt werden soll. In der Regel umfasst ein Workflow mehrere Aktionen, die nacheinander oder parallel ausgeführt werden, je nachdem, wie Sie sie konfiguriert haben.

Weitere Informationen zu Aktionen finden Sie unter [Verwenden von Aktionen](#).

## Aktionsgruppen

Eine Aktionsgruppe enthält eine oder mehrere Aktionen. Das Gruppieren von Aktionen in Aktionsgruppen hilft Ihnen dabei, Ihren Arbeitsablauf zu organisieren, und ermöglicht es Ihnen auch, Abhängigkeiten zwischen verschiedenen Aktionsgruppen zu konfigurieren.

Weitere Informationen zu Aktionsgruppen finden Sie unter [Gruppierung von Aktionen in Aktionsgruppen](#).

## -Artefakte

Ein Artefakt ist das Ergebnis einer Workflow-Aktion und besteht in der Regel aus einem Ordner oder Archiv mit Dateien. Artefakte sind wichtig, weil sie es Ihnen ermöglichen, Dateien und Informationen zwischen Aktionen gemeinsam zu nutzen.

Weitere Informationen zu Artefakten finden Sie unter [Mit Artefakten arbeiten](#).

## Datenverarbeitung

Compute bezieht sich auf die Computing-Engine (CPU, Arbeitsspeicher und Betriebssystem), die CodeCatalyst zur Ausführung von Workflow-Aktionen verwaltet und gewartet wird.

Weitere Informationen zu Compute finden Sie unter [Mit Compute arbeiten](#).

## Umgebungen

In einer Umgebung, nicht zu verwechseln mit einer [Entwicklungsumgebung](#), wird Code bereitgestellt. Sie enthält normalerweise eine Instanz einer laufenden Anwendung zusammen mit der zugehörigen Infrastruktur. Sie können Ihrer Umgebung einen Namen wie Entwicklung, Test, Staging oder Produktion geben. Alle Bereitstellungen, die von einer Umgebung generiert wurden CodeCatalyst, werden auf der Seite Umgebungen angezeigt. Um eine Umgebung einzurichten, geben Sie ihr einen Namen, z. B. `my-production-environment`, und verknüpfen sie dann mit Ihrer AWS-Konto.

Neben der Anzeige von Bereitstellungsinformationen dient eine Umgebung auch als Mechanismus für die Zuweisung von AWS IAM-Rollen zu [Workflow-Aktionen](#).

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#).

## Berichte

Ein Bericht enthält Details zu Tests, die während einer Workflow-Ausführung durchgeführt werden. Sie können Berichte wie einen Testbericht, einen Bericht zur Codeabdeckung, einen Analysebericht zur Softwarezusammensetzung und einen statischen Analysebericht erstellen. Sie können einen Bericht verwenden, um ein Problem während eines Workflows zu beheben. Wenn Sie über viele Berichte aus mehreren Workflows verfügen, können Sie Ihre Berichte verwenden, um Trends und Ausfallraten zu ermitteln und so Ihre Anwendungen und Bereitstellungsconfigurationen zu optimieren.

Weitere Informationen zu Berichten finden Sie unter [Typen von Testberichten](#).

## Ausführungen

Ein Lauf ist eine einzelne Iteration eines Workflows. CodeCatalyst führt während eines Laufs die in der Workflow-Konfigurationsdatei definierten Aktionen aus und gibt die zugehörigen Protokolle, Artefakte und Variablen aus.

Weitere Informationen zu Läufen finden Sie unter [Mit Läufen arbeiten](#).

## Quellen

Eine Quelle, auch Eingabequelle genannt, ist ein Quell-Repository, auf das eine [Workflow-Aktion](#) Zugriff benötigt, um ihre Aufgaben auszuführen. Beispielsweise muss eine Workflow-Aktion möglicherweise auf eine Quelle zugreifen, um Komponententests abzurufen und sie anhand der Quelldateien Ihrer Anwendung auszuführen.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

## Variablen

Eine Variable ist ein Schlüssel-Wert-Paar, das Informationen enthält, auf die Sie in Ihrem CodeCatalyst Workflow verweisen können.

Weitere Informationen zu Variablen finden Sie unter [Arbeiten mit Variablen](#).

## Workflow-Auslöser

Mit einem Workflow-Trigger oder einfach einem Trigger können Sie eine Workflow-Ausführung automatisch starten, wenn bestimmte Ereignisse eintreten, z. B. ein Code-Push. Möglicherweise möchten Sie Trigger so konfigurieren, dass Ihre Softwareentwickler Workflow-Läufe nicht manuell über die CodeCatalyst Konsole starten müssen.

Sie können drei Arten von Triggern verwenden:

- Push — Ein Code-Push-Trigger bewirkt, dass ein Workflow-Lauf immer dann gestartet wird, wenn ein Commit übertragen wird.
- Pull-Request — Ein Pull-Request-Trigger bewirkt, dass ein Workflow-Lauf immer dann gestartet wird, wenn ein Pull-Request entweder erstellt, überarbeitet oder geschlossen wird.
- Zeitplan — Ein Zeitplan-Trigger bewirkt, dass ein Workflow-Lauf nach einem von Ihnen definierten Zeitplan gestartet wird. Erwägen Sie, einen Zeitplan-Trigger zu verwenden, um nächtliche Builds

Ihrer Software auszuführen, sodass Ihre Softwareentwickler am nächsten Morgen mit der neuesten Version arbeiten können.

Sie können Push-, Pull-Request- und Schedule-Trigger einzeln oder in Kombination im selben Workflow verwenden.

Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).

## Erste Schritte mit Workflows in CodeCatalyst

In diesem Tutorial erfahren Sie, wie Sie Ihren ersten Workflow erstellen und konfigurieren.

### Tip

Möchten Sie lieber mit einem vorkonfigurierten Workflow beginnen? Siehe [Ein Projekt mit einem Blueprint erstellen](#), dort finden Sie Anweisungen zum Einrichten eines Projekts mit einem funktionierenden Workflow, eine Beispielanwendung und andere Ressourcen.

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen und konfigurieren Sie Ihren Workflow](#)
- [Schritt 2: Speichern Sie Ihren Workflow mit einem Commit](#)
- [Schritt 3: Ausführungsergebnisse anzeigen](#)
- [\(Optional\) Schritt 4: Bereinigen](#)

## Voraussetzungen

Bevor Sie beginnen:

- Sie benötigen ein CodeCatalyst Leerzeichen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).
- In Ihrem CodeCatalyst Bereich benötigen Sie ein leeres CodeCatalyst Projekt mit dem Namen:

```
codecatalyst-project
```



Weitere Informationen finden Sie unter [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#).

- In Ihrem Projekt benötigen Sie ein CodeCatalyst Repository mit dem Namen:

```
codecatalyst-source-repository
```

Weitere Informationen finden Sie unter [Erstellen eines Quell-Repositorys](#).

#### Note

Wenn Sie bereits über ein Projekt und ein Quell-Repository verfügen, können Sie diese verwenden. Wenn Sie jedoch neue erstellen, wird die Bereinigung am Ende dieses Tutorials erleichtert.

## Schritt 1: Erstellen und konfigurieren Sie Ihren Workflow

In diesem Schritt erstellen und konfigurieren Sie einen Workflow, der Ihren Quellcode automatisch erstellt und testet, wenn Änderungen vorgenommen werden.

Um Ihren Workflow zu erstellen

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie Workflow erstellen aus.

Die Workflow-Definitionsdatei wird im YAML-Editor der CodeCatalyst Konsole angezeigt.

Um Ihren Workflow zu konfigurieren

Sie können Ihren Workflow im Visual Editor oder im YAML-Editor konfigurieren. Beginnen wir mit dem YAML-Editor und wechseln dann zum visuellen Editor.

1. Wählen Sie + Aktionen, um eine Liste von Workflow-Aktionen anzuzeigen, die Sie Ihrem Workflow hinzufügen können.
2. Wählen Sie in der Aktion Erstellen + aus, um das YAML der Aktion zu Ihrer Workflow-Definitionsdatei hinzuzufügen. Ihr Workflow sieht jetzt wie folgt aus.

```
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
        - Run: echo "<testsuite tests=\"1\" name=\"TestAgentJunit\" >" >>
report.xml
        - Run: echo "<testcase classname=\"TestAgentJunit\" name=\"Dummy
Test\"/></testsuite>" >> report.xml
```

Der Workflow kopiert die Dateien im **WorkflowSource** Quell-Repository auf den Computer, auf dem die **Build\_f0** Aktion ausgeführt wird, druckt **Hello, World!** in die Protokolle, erkennt Testberichte auf dem Computercomputer und gibt sie auf der Berichtsseite der CodeCatalyst Konsole aus.

3. Wählen Sie Visual, um die Workflow-Definitionsdatei im Visual Editor anzuzeigen. Mit den Feldern im visuellen Editor können Sie die im YAML-Editor angezeigten YAML-Eigenschaften konfigurieren.

## Schritt 2: Speichern Sie Ihren Workflow mit einem Commit

In diesem Schritt speichern Sie Ihre Änderungen. Da Workflows als `.yaml` Dateien in Ihrem Repository gespeichert werden, speichern Sie Ihre Änderungen mit Commits.

Um Ihre Workflow-Änderungen zu übernehmen

1. (Optional) Wählen Sie Validieren, um sicherzustellen, dass der YAML-Code des Workflows gültig ist.
2. Wählen Sie Commit (Übergeben).
3. Geben Sie im Feld Workflow-Dateiname einen Namen für Ihre Workflow-Konfigurationsdatei ein, z. B. **my-first-workflow**
4. Geben Sie im Feld Commit-Nachricht eine Nachricht ein, um Ihren Commit zu identifizieren, z. **create my-first-workflow.yaml** B.
5. Wählen Sie unter Repository das Repository aus, in dem Sie den Workflow speichern möchten (`codecatalyst-repository`).
6. Wählen Sie unter Branchname den Branch aus, in dem Sie den Workflow speichern möchten (`main`).
7. Wählen Sie Commit (Übergeben).

Ihr neuer Workflow wird in der Liste der Workflows angezeigt. Es kann einen Moment dauern, bis es angezeigt wird.

Da Workflows mit Commits gespeichert werden und für den Workflow ein Code-Push-Trigger konfiguriert ist, wird beim Speichern des Workflows automatisch eine Workflow-Ausführung gestartet.

## Schritt 3: Ausführungsergebnisse anzeigen

In diesem Schritt navigieren Sie zu dem Lauf, der von Ihrem Commit aus gestartet wurde, und sehen sich die Ergebnisse an.

Um die Ergebnisse des Laufs anzuzeigen

1. Wählen Sie den Namen Ihres Workflows, zum Beispiel `Workflow_fe47`.

Ein Workflow-Diagramm, das die Bezeichnung Ihres Quell-Repositorys (`WorkflowSource`) und die Build-Aktion (z. B. `build_F0`) zeigt.

2. Wählen Sie im Workflow-Ausführungsdigramm die Build-Aktion aus (z. B. `build_F0`).

- Überprüfen Sie den Inhalt der Registerkarten Protokolle, Berichte, Konfiguration und Variablen. Auf diesen Registerkarten werden die Ergebnisse Ihrer Build-Aktion angezeigt.

Weitere Informationen finden Sie unter [Ergebnisse einer Build-Aktion anzeigen](#).

## (Optional) Schritt 4: Bereinigen

In diesem Schritt bereinigen Sie die Ressourcen, die Sie in diesem Tutorial erstellt haben.

Um Ressourcen zu löschen

- Wenn Sie für dieses Tutorial ein neues Projekt erstellt haben, löschen Sie es. Anweisungen finden Sie unter [Löschen eines Projekts in Amazon CodeCatalyst](#). Durch das Löschen des Projekts werden auch das Quell-Repository und der Workflow gelöscht.
- Löschen Sie den Workflow, falls er noch nicht gelöscht wurde. Anweisungen finden Sie unter [Um einen Workflow zu löschen](#).
- Löschen Sie das Repository, falls es noch nicht gelöscht wurde. Detaillierte Anweisungen finden Sie unter [Löschen eines Quell-Repositorys](#).

## Codequalität und Bereitstellungsstatus nach Commit in anzeigen CodeCatalyst

Zu jedem Zeitpunkt im Entwicklungszyklus ist es wichtig, den Bereitstellungsstatus bestimmter Commits zu kennen, z. B. Bugfixes, neue Funktionen oder andere wichtige Änderungen. Stellen Sie sich die folgenden Szenarien vor, in denen die Fähigkeit zur Nachverfolgung des Bereitstellungsstatus für Entwicklungsteams hilfreich ist:

- Als Entwickler haben Sie einen Fehler behoben und möchten den Status der Implementierung in den Bereitstellungsumgebungen Ihres Teams melden.
- Als Release-Manager möchten Sie eine Liste der bereitgestellten Commits einsehen, um deren Bereitstellungsstatus nachzuverfolgen und zu melden.

CodeCatalyst bietet eine Ansicht, anhand derer Sie auf einen Blick feststellen können, wo und in welcher Umgebung einzelne Commits oder Änderungen implementiert wurden. Diese Ansicht beinhaltet:

- Eine Liste von Commits.
- Der Status der Bereitstellungen, die die Commits enthalten.
- Die Umgebungen, in denen die Commits erfolgreich bereitgestellt wurden.
- Der Status aller Tests, die anhand der Commits in Ihrem CI/CD-Workflow ausgeführt wurden.

Das folgende Verfahren beschreibt, wie Sie zu dieser Ansicht navigieren und sie verwenden, um Änderungen in Ihrem Projekt nachzuverfolgen.

#### Note

Das Verfolgen des Bereitstellungsstatus per Commit wird nur mit [CodeCatalyst Repositorys](#) unterstützt. Sie können diese Funktion nicht mit einem [GitHub Repository](#) verwenden.

Um den Bereitstellungsstatus per Commit zu verfolgen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Change Tracking aus.
4. Wählen Sie in den beiden Dropdownlisten oben im Hauptfenster das Quell-Repository und den Branch aus, die die Commits enthalten, deren Release-Status Sie anzeigen möchten.
5. Wähle „Änderungen anzeigen“.

Eine Liste mit Commits wird angezeigt.

Für jeden Commit können Sie Folgendes einsehen:

- Commit-Informationen wie ID, Autor, Nachricht und Datum der Übertragung. Weitere Informationen finden Sie unter [Quell-Repositoryn in CodeCatalyst](#).
- Der Status der Bereitstellungen in jeder Umgebung. Weitere Informationen finden Sie unter [Arbeiten mit Umgebungen](#).
- Ergebnisse der Tests und der Codeabdeckung. Weitere Informationen finden Sie unter [Testen mit Workflows in CodeCatalyst](#).

**Note**

Die Ergebnisse der Software Composition Analysis (SCA) werden nicht angezeigt.

6. (Optional) Um weitere Informationen zu den Änderungen im Zusammenhang mit einem bestimmten Commit anzuzeigen, einschließlich der neuesten Bereitstellung und detaillierter Informationen zur Codeabdeckung und zum Komponententest, wählen Sie Details für diesen Commit anzuzeigen.

## Erstellen mithilfe von Workflows in CodeCatalyst

Mithilfe von [CodeCatalyst Workflows](#) können Sie Anwendungen und andere Ressourcen erstellen.

### Themen

- [Wie erstelle ich eine Anwendung?](#)
- [Vorteile der Build-Aktion](#)
- [Alternativen zur Build-Aktion](#)
- [Hinzufügen der Build-Aktion](#)
- [Ergebnisse einer Build-Aktion anzeigen](#)
- [Tutorial: Artefakte auf Amazon S3 hochladen](#)

## Wie erstelle ich eine Anwendung?

Um eine Anwendung oder Ressource darin zu erstellen CodeCatalyst, erstellen Sie zunächst einen Workflow und geben dann eine darin enthaltene Build-Aktion an.

Eine Build-Aktion ist ein Workflow-Baustein, der Ihren Quellcode kompiliert, Komponententests ausführt und Artefakte erzeugt, die sofort bereitgestellt werden können.

Sie fügen Ihrem Workflow mithilfe des Visual Editors oder YAML-Editors der CodeCatalyst Konsole eine Build-Aktion hinzu.

Die allgemeinen Schritte zum Erstellen einer Anwendung oder Ressource lauten wie folgt.

## Um eine Anwendung zu erstellen (Aufgaben auf hoher Ebene)

1. In CodeCatalyst fügen Sie Quellcode für eine Anwendung hinzu, die Sie erstellen möchten. Weitere Informationen finden Sie unter [Arbeiten mit Quell-Repositorys in CodeCatalyst](#).
2. CodeCatalystIn erstellen Sie einen Workflow. In diesem Workflow definieren Sie, wie Ihre Anwendung erstellt, getestet und bereitgestellt werden soll. Weitere Informationen finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#).
3. (Optional) Im Workflow fügen Sie einen Trigger hinzu, der die Ereignisse angibt, die dazu führen, dass der Workflow automatisch gestartet wird. Weitere Informationen finden Sie unter [Arbeiten mit Auslösern](#).
4. Im Workflow fügen Sie eine Build-Aktion hinzu, die den Quellcode Ihrer Anwendung oder Ressource kompiliert und verpackt. Optional können Sie mit der Build-Aktion auch Komponententests ausführen, Berichte generieren und Ihre Anwendung bereitstellen, wenn Sie für diese Zwecke keine Test- oder Bereitstellungsaktion verwenden möchten. Weitere Informationen zu den Test- und Bereitstellungsaktionen finden Sie unter [Hinzufügen der Build-Aktion](#).
5. (Optional) Im Workflow fügen Sie eine Testaktion und eine Bereitstellungsaktion hinzu, um Ihre Anwendung oder Ressource zu testen und bereitzustellen. Sie können aus mehreren vorkonfigurierten Aktionen wählen, um Ihre Anwendung für verschiedene Ziele bereitzustellen, z. B. Amazon ECS. Weitere Informationen finden Sie unter [Testen mit Workflows in CodeCatalyst](#) und [Bereitstellung mithilfe von Workflows in CodeCatalyst](#).
6. Sie starten den Workflow entweder manuell oder automatisch über einen Trigger. Der Workflow führt die Build-, Test- und Bereitstellungsaktionen nacheinander aus, um Ihre Anwendung und Ressourcen zu erstellen, zu testen und auf dem Ziel bereitzustellen. Weitere Informationen finden Sie unter [Einen Workflow-Lauf starten](#).

## Vorteile der Build-Aktion

Die Verwendung der Build-Aktion innerhalb eines Workflows hat die folgenden Vorteile:

- Vollständig verwaltet — Durch die Build-Aktion müssen Sie Ihre eigenen Build-Server nicht mehr einrichten, patchen, aktualisieren und verwalten.
- Bei Bedarf — Die Build-Aktion wird nach Bedarf skaliert, um Ihren Build-Anforderungen gerecht zu werden. Sie zahlen nur für die Anzahl der Build-Minuten, die Sie wirklich nutzen. Weitere Informationen finden Sie unter [Mit Compute arbeiten](#).

- **Sofort einsatzbereit** — CodeCatalyst enthält vorkonfigurierte Docker-Images für die Laufzeitumgebung, die zur Ausführung all Ihrer Workflow-Aktionen, einschließlich Build-Aktionen, verwendet werden. Diese Images sind mit nützlichen Tools für die Erstellung von Anwendungen wie Node.js und The AWS CLI vorkonfiguriert. Sie können so konfigurieren CodeCatalyst , dass ein Build-Image verwendet wird, das Sie aus einer öffentlichen oder privaten Registrierung bereitstellen. Weitere Informationen finden Sie unter [Arbeiten mit Docker-Images für die Laufzeitumgebung](#).

## Alternativen zur Build-Aktion

Wenn Sie eine Build-Aktion verwenden, um Ihre Anwendung bereitzustellen, sollten Sie stattdessen eine CodeCatalyst Bereitstellungsaktion verwenden. Bereitstellungsaktionen führen behind-the-scenes Konfigurationen durch, die Sie andernfalls manuell schreiben müssten, wenn Sie eine Build-Aktion verwenden würden. Weitere Informationen zu den verfügbaren Bereitstellungsaktionen finden Sie unter [Liste der Bereitstellungsaktionen](#).

Sie können es auch verwenden AWS CodeBuild , um Ihre Anwendungen zu erstellen. Weitere Informationen finden Sie unter [Was ist CodeBuild?](#) .

### Themen

- [Hinzufügen der Build-Aktion](#)
- [Ergebnisse einer Build-Aktion anzeigen](#)
- [Tutorial: Artefakte auf Amazon S3 hochladen](#)

## Hinzufügen der Build-Aktion

Gehen Sie wie folgt vor, um Ihrem CodeCatalyst Workflow eine Build-Aktion hinzuzufügen.

### Visual

Um eine Build-Aktion mit dem Visual Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.



4. Wählen Sie Bearbeiten aus.
5. Wählen Sie Visual.
6. Wählen Sie Aktionen.
7. Wählen Sie unter Aktionen die Option Build aus.
8. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Aktionsreferenz zum Erstellen und Testen](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um eine Build-Aktion mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
4. Wählen Sie Bearbeiten aus.
5. Wählen Sie YAML.
6. Wählen Sie Aktionen.
7. Wählen Sie unter Aktionen die Option Build aus.
8. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Aktionsreferenz zum Erstellen und Testen](#).
9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Erstellen Sie eine Aktionsdefinition

Die Build-Aktion ist als ein Satz von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Aktionsreferenz zum Erstellen und Testen](#) in der [Referenz zur Workflow-Definition](#).

## Ergebnisse einer Build-Aktion anzeigen

Verwenden Sie die folgenden Anweisungen, um die Ergebnisse einer Build-Aktion, einschließlich der generierten Protokolle, Berichte und Variablen, anzuzeigen.

So zeigen Sie die Ergebnisse einer Build-Aktion an

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
3. Wählen Sie im Workflow-Diagramm den Namen Ihrer Build-Aktion aus, zum Beispiel Build.
4. Um die Protokolle für den Build-Lauf anzuzeigen, wählen Sie Logs aus. Die Logs für die verschiedenen Build-Phasen werden angezeigt. Sie können die Protokolle nach Bedarf erweitern.
5. Um die durch die Build-Aktion erstellten Testberichte anzuzeigen, wählen Sie Berichte aus, oder wählen Sie im Navigationsbereich Berichte aus. Weitere Informationen finden Sie unter [Typen von Testberichten](#).
6. Um die für die Build-Aktion verwendete Konfiguration anzuzeigen, wählen Sie Konfiguration aus. Weitere Informationen finden Sie unter [Hinzufügen der Build-Aktion](#).
7. Um die von der Build-Aktion verwendeten Variablen anzuzeigen, wählen Sie Variablen. Weitere Informationen finden Sie unter [Arbeiten mit Variablen](#).

## Tutorial: Artefakte auf Amazon S3 hochladen

In diesem Tutorial erfahren Sie, wie Sie mithilfe eines CodeCatalyst [Workflows](#), der einige [Build-Aktionen](#) umfasst, Artefakte in einen Amazon S3 S3-Bucket hochladen. Diese Aktionen werden nacheinander ausgeführt, wenn der Workflow gestartet wird. Die erste Build-Aktion generiert zwei Dateien `Hello.txt` und `Goodbye.txt`, und bündelt sie zu einem Build-Artefakt. Die zweite Build-Aktion lädt das Artefakt auf Amazon S3 hoch. Sie konfigurieren den Workflow so, dass er jedes Mal ausgeführt wird, wenn Sie einen Commit in Ihr Quell-Repository übertragen.

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie eine AWS Rolle](#)
- [Schritt 2: Erstellen Sie einen Amazon S3 S3-Bucket](#)
- [Schritt 3: Erstellen Sie ein Quell-Repository](#)
- [Schritt 4: Erstellen Sie einen Workflow](#)
- [Schritt 5: Überprüfen Sie die Ergebnisse](#)
- [Bereinigen](#)

## Voraussetzungen

Bevor Sie beginnen, muss Folgendes sichergestellt sein:

- Sie benötigen einen CodeCatalyst Bereich mit einem verbundenen AWS Konto. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).
- In Ihrem Bereich benötigen Sie ein leeres CodeCatalyst Projekt mit dem Namen „Von vorne beginnen“:

```
codecatalyst-artifact-project
```

Weitere Informationen finden Sie unter [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#).

- In Ihrem Projekt benötigen Sie eine CodeCatalyst Umgebung namens:

```
codecatalyst-artifact-environment
```

Konfigurieren Sie diese Umgebung wie folgt:

- Wählen Sie einen beliebigen Typ, z. B. Entwicklung.
- Connect dein AWS Konto damit.

Weitere Informationen finden Sie unter [Arbeiten mit Umgebungen](#).

## Schritt 1: Erstellen Sie eine AWS Rolle

In diesem Schritt erstellen Sie eine AWS IAM-Rolle, die Sie später der Build-Aktion in Ihrem Workflow zuweisen. Diese Rolle gewährt der CodeCatalyst Build-Aktion die Berechtigung, auf Ihr AWS Konto

zugreifen und in Amazon S3 zu schreiben, wo Ihr Artefakt gespeichert wird. Die Rolle wird Build-Rolle genannt.

### Note

Wenn Sie bereits über eine Build-Rolle verfügen, die Sie für ein anderes Tutorial erstellt haben, können Sie sie auch für dieses Tutorial verwenden. Stellen Sie einfach sicher, dass sie über die im folgenden Verfahren beschriebenen Berechtigungen und Vertrauensrichtlinien verfügt.

Weitere Informationen zu IAM-Rollen finden Sie unter [IAM-Rollen](#) im AWS Identity and Access Management Benutzerhandbuch.

Um eine Build-Rolle zu erstellen

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich an bei AWS.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
  - c. Wählen Sie im Navigationsbereich Policies aus.
  - d. Wählen Sie Richtlinie erstellen aus.
  - e. Wählen Sie den Tab JSON.
  - f. Löschen Sie den vorhandenen Code.
  - g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

}

**Note**

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie im Feld Name Folgendes ein:

```
codecatalyst-s3-build-policy

```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

## 2. Erstellen Sie die Build-Rolle wie folgt:

- a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie.
- c. Löschen Sie die bestehende benutzerdefinierte Vertrauensrichtlinie.
- d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      }
    }
  ]
}
```

```
        },
        "Action": "sts:AssumeRole"
    }
]
}
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach dem entsprechenden Kontrollkästchen `codecatalyst-s3-build-policy` und aktivieren Sie es.
- g. Wählen Sie Weiter aus.
- h. Geben Sie als Rollenname Folgendes ein:

```
codecatalyst-s3-build-role
```

- i. Geben Sie als Rollenbeschreibung Folgendes ein:

```
CodeCatalyst build role
```

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

## Schritt 2: Erstellen Sie einen Amazon S3 S3-Bucket

In diesem Schritt erstellen Sie einen Amazon S3 S3-Bucket, in den die `Hello.txt` und `Goodbye.txt` -Artefakte hochgeladen werden.

So erstellen Sie einen Amazon-S3-Bucket

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Hauptbereich Bucket erstellen aus.
3. Geben Sie als Bucket-Namen Folgendes ein:

```
codecatalyst-artifact-bucket
```

4. Wählen Sie unter AWS -Region eine Region aus. In diesem Tutorial wird davon ausgegangen, dass Sie US West (Oregon) `us-west-2` ausgewählt haben. Informationen zu den von Amazon

S3 unterstützten Regionen finden Sie unter [Amazon Simple Storage Service-Endpunkte und Kontingente](#) in der Allgemeine AWS-Referenz.

5. Wählen Sie unten auf der Seite die Option Bucket erstellen aus.
6. Kopieren Sie den Namen des Buckets, den Sie gerade erstellt haben, zum Beispiel:

```
codecatalyst-artifact-bucket
```

Sie haben jetzt einen Bucket mit dem Namen **codecatalyst-artifact-bucket** in der Region USA West (Oregon) us-west-2 erstellt.

### Schritt 3: Erstellen Sie ein Quell-Repository

In diesem Schritt erstellen Sie ein Quell-Repository in CodeCatalyst. Dieses Repository wird verwendet, um die Workflow-Definitionsdatei des Tutorials zu speichern.

Weitere Informationen zu Quell-Repositorys finden Sie unter [Erstellen eines Quell-Repositorys](#).

Um ein Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt, `codecatalyst-artifact-project`.
3. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.
4. Wählen Sie Repository hinzufügen und anschließend Repository erstellen aus.
5. Geben Sie im Feld Repository-Name Folgendes ein:

```
codecatalyst-artifact-source-repository
```

6. Wählen Sie Erstellen.

Sie haben jetzt ein Repository mit dem Namen `codecatalyst-artifact-source-repository`.

### Schritt 4: Erstellen Sie einen Workflow

In diesem Schritt erstellen Sie einen Workflow, der aus den folgenden Bausteinen besteht, die nacheinander ausgeführt werden:

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Triggern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine Build-Aktion namens GenerateFiles — Beim Trigger erstellt die GenerateFiles Aktion zwei Dateien Hello.txt und Goodbye.txt packt sie in ein Ausgabeartefakt namens codecatalystArtifact.
- Eine weitere Build-Aktion namens Upload — Nach Abschluss der GenerateFiles Aktion führt die Upload Aktion den AWS CLI Befehl aus, `aws s3 sync` um die Dateien im codecatalystArtifact und in Ihrem Quell-Repository in Ihren Amazon S3 S3-Bucket hochzuladen. Das AWS CLI ist auf der CodeCatalyst Rechenplattform vorinstalliert und vorkonfiguriert, sodass Sie es nicht installieren oder konfigurieren müssen.

Weitere Informationen zur vorkonfigurierten Software auf der CodeCatalyst Rechenplattform finden Sie unter [Arbeiten mit Docker-Images für die Laufzeitumgebung](#) Weitere Informationen zum `aws s3 sync` Befehl AWS CLI's finden Sie unter [sync](#) in der AWS CLI Befehlsreferenz.

Weitere Informationen zur Build-Aktion finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#).

So erstellen Sie ein Workflow

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie Workflow erstellen aus.
3. Löschen Sie den YAML-Beispielcode.
4. Fügen Sie den folgenden YAML-Code hinzu:

```
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
```

Ersetzen Sie im obigen Code:

- *codecatalyst-artifact-environment* durch den Namen der Umgebung, in der Sie erstellt haben [Voraussetzungen](#).
- *codecatalyst-account-connection* mit dem Namen der Kontoverbindung, in der Sie eine Verbindung erstellt haben [Voraussetzungen](#).
- *codecatalyst-s3-build-role mit dem Namen der Build-Rolle*, in der Sie erstellt haben. [Schritt 1: Erstellen Sie eine AWS Rolle](#)
- *codecatalyst-artifact-bucket* mit dem Namen des Amazon S3, in dem Sie es erstellt haben [Schritt 2: Erstellen Sie einen Amazon S3 S3-Bucket](#).

Informationen zu den Eigenschaften in dieser Datei finden Sie unter [Aktionsreferenz zum Erstellen und Testen](#).

5. (Optional) Wählen Sie „Validieren“, um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie ihn festschreiben.

6. Wählen Sie Commit (Übergeben).
7. Geben Sie im Dialogfeld „Workflow bestätigen“ Folgendes ein:
  - a. Behalten Sie für Workflow-Dateiname die Standardeinstellung `beibodecatalyst-artifact-workflow`.
  - b. Geben Sie für Commit-Nachricht Folgendes ein:

**add initial workflow file**

- c. Wählen Sie für Repository `codecatalyst-artifact-source-repository`.
- d. Wählen Sie als Branch-Name die Option `main` aus.
- e. Wählen Sie Commit (Übergeben).

Sie haben jetzt einen Workflow erstellt. Eine Workflow-Ausführung wird aufgrund des oben im Workflow definierten Triggers automatisch gestartet. Insbesondere wenn Sie die `codecatalyst-artifact-workflow.yaml` Datei in Ihr Quell-Repository übertragen (und per Push) haben, hat der Trigger die Workflow-Ausführung gestartet.

Um den laufenden Workflow-Lauf zu sehen

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie den Workflow aus, den Sie gerade erstellt haben: `codecatalyst-artifact-workflow`
3. Wählen Sie `GenerateFiles`, ob Sie den Fortschritt der ersten Build-Aktion sehen möchten.
4. Wählen Sie `Hochladen`, um den Fortschritt der zweiten Build-Aktion zu sehen.
5. Wenn die Upload-Aktion abgeschlossen ist, gehen Sie wie folgt vor:
  - Wenn die Workflow-Ausführung erfolgreich war, fahren Sie mit dem nächsten Verfahren fort.
  - Wenn die Workflow-Ausführung fehlgeschlagen ist, wählen Sie `Protokolle` aus, um das Problem zu beheben.

## Schritt 5: Überprüfen Sie die Ergebnisse

Nachdem der Workflow ausgeführt wurde, rufen Sie den Amazon S3 S3-Service auf und schauen Sie in Ihrem `codecatalyst-artifact-bucket` Bucket nach. Es sollte jetzt die folgenden Dateien und Ordner enthalten:

```
.
|- .aws/
|- .git/
|Goodbye.txt
|Hello.txt
|README.md
```

Die `Hello.txt` Dateien `Goodbye.txt` und wurden hochgeladen, weil sie Teil des `codecatalystArtifact` Artefakts waren. Die `README.md` Dateien `.aws/` `.git/`, und wurden hochgeladen, weil sie sich in Ihrem Quell-Repository befanden.

## Bereinigen

Räumen Sie auf CodeCatalyst und vermeiden Sie AWS , dass Ihnen diese Dienste in Rechnung gestellt werden.

Zum Aufräumen in CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Löschen Sie das `codecatalyst-artifact-source-repository` Quell-Repository.
3. Löschen Sie den `codecatalyst-artifact-workflow` Workflow.

Zum Aufräumen in AWS

1. Bereinigen Sie in Amazon S3 wie folgt:
  - a. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
  - b. Löschen Sie die Dateien im `codecatalyst-artifact-bucket` Bucket.
  - c. Löschen Sie den `codecatalyst-artifact-bucket` Bucket.
2. Bereinigen Sie in IAM wie folgt:
  - a. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.

- b. Löschen Sie das `codecatalyst-s3-build-policy`.
- c. Löschen Sie das `codecatalyst-s3-build-role`.

## Testen mit Workflows in CodeCatalyst

CodeCatalystIn können Sie Tests als Teil verschiedener Workflow-Aktionen wie Erstellen und Testen ausführen. Diese Workflow-Aktionen können alle Berichte generieren. Eine Testaktion ist eine Workflow-Aktion, mit der Berichte zu Tests, Codeabdeckung, Softwarekomposition und statischen Analysen erstellt werden. Diese Berichte werden in der CodeCatalyst Konsole angezeigt.

### Themen

- [Typen von Testberichten](#)
- [Testaktion hinzufügen](#)
- [Berichte konfigurieren](#)
- [Benutzen universal-test-runner](#)
- [Bewährte Methoden](#)
- [Mit Tests arbeiten](#)

## Typen von Testberichten

Die CodeCatalyst Amazon-Testaktion unterstützt die folgenden Berichtstypen. Ein Beispiel zur Formatierung dieser Berichte in Ihrem YAML finden Sie unter [Beispiel: Berichte konfigurieren](#).

### Themen

- [Testberichte](#)
- [Berichte zur Codeabdeckung](#)
- [Berichte zur Analyse der Softwarezusammensetzung](#)
- [Statische Analyseberichte](#)

## Testberichte

In CodeCatalyst können Sie Komponententests, Integrationstests und Systemtests konfigurieren, die während Builds ausgeführt werden. Anschließend CodeCatalyst können Sie Berichte erstellen, die die Ergebnisse Ihrer Tests enthalten.

Sie können einen Testbericht verwenden, um Probleme mit Ihren Tests zu beheben. Wenn Sie über viele Testberichte von mehreren Builds verfügen, können Sie anhand Ihrer Testberichte die Fehlerraten anzeigen und so Ihre Builds optimieren.

Sie können die folgenden Dateiformate für Testberichte verwenden:

- Gurke JSON (.json)
- Einheits-XML (.xml)
- Einheit XML (.xml)
- NUnit3-XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx, .xml)

## Berichte zur Codeabdeckung

CodeCatalystIn können Sie Berichte zur Codeabdeckung für Ihre Tests erstellen. CodeCatalyst bietet die folgenden Kennzahlen zur Codeabdeckung:

### Leitungsabdeckung

Misst, wie viele Aussagen Ihre Tests abdecken. Eine Aussage ist eine einzelne Anweisung, ohne Kommentare.

```
line coverage = (total lines covered)/(total number of lines)
```

### Abdeckung der Filialen

Misst, wie viele Zweige Ihre Tests von allen möglichen Zweigen einer Kontrollstruktur, wie z. B. einer if case Oder-Anweisung, abdecken.

```
branch coverage = (total branches covered)/(total number of branches)
```

Die folgenden Dateiformate für Berichte zur Codeabdeckung werden unterstützt:

- JaCoCo XML (.xml)
- SimpleCov [JSON \(generiert von simplecov, nicht simplecov-json, .json\)](#)
- Clover XML (Version 3, .xml)
- XML-Cover (.xml)

- LCOV (.info)

## Berichte zur Analyse der Softwarezusammensetzung

In CodeCatalyst können Sie Tools zur Software Composition Analysis (SCA) verwenden, um Komponenten Ihrer Anwendung zu analysieren und nach bekannten Sicherheitslücken zu suchen. Sie können SARIF-Berichte entdecken und analysieren, in denen Sicherheitslücken mit unterschiedlichem Schweregrad und Möglichkeiten zu ihrer Behebung detailliert beschrieben werden. Gültige Schweregradwerte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICAL,,HIGH,MEDIUM. LOW INFORMATIONAL

Die folgenden Dateiformate für SCA-Berichte werden unterstützt:

- SARIF (.sarif, .json)

## Statische Analyseberichte

Sie können statische Analyseberichte (SA) verwenden, um Codefehler auf Quelltextebene zu identifizieren. In können Sie SA-Berichte generieren CodeCatalyst, um Probleme in Ihrem Code zu beheben, bevor Sie ihn bereitstellen. Zu diesen Problemen gehören Bugs, Sicherheitslücken, Qualitätsprobleme und andere Sicherheitslücken. Gültige Schweregradwerte, vom höchsten bis zum niedrigsten Schweregrad CRITICAL, sind: HIGHMEDIUM,LOW,, undINFORMATIONAL.

CodeCatalyst bietet die folgenden SA-Metriken:

### Bugs

Identifiziert eine Reihe möglicher Fehler in Ihrem Quellcode. Diese Fehler können Probleme mit der Speichersicherheit beinhalten. Das Folgende ist ein Beispiel für einen Fehler.

```
// 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;
}
```

## Sicherheitslücken

Identifiziert eine Reihe möglicher Sicherheitslücken in Ihrem Quellcode. Zu diesen Sicherheitslücken können Probleme wie das Speichern Ihrer geheimen Token im Klartext gehören.

## Probleme mit der Qualität

Identifiziert eine Reihe möglicher Qualitätsprobleme in Ihrem Quellcode. Zu diesen Qualitätsproblemen können auch Probleme im Zusammenhang mit Stilkonventionen gehören. Im Folgenden finden Sie ein Beispiel für ein Qualitätsproblem.

```
// The function name doesn't adhere to the style convention of camelCase
int SUBTRACT(int x, int y) {
    return x-y
}
```

## Andere Sicherheitslücken

Identifiziert eine Reihe möglicher anderer Sicherheitslücken, die in Ihrem Quellcode gefunden wurden.

CodeCatalyst unterstützt die folgenden SA-Berichtsdateiformate:

- PyLint (.py)
- ESLint (.js, .jsx, .ts, .tsx)
- SARIF (.sarif, .json)

## Testaktion hinzufügen

Gehen Sie wie folgt vor, um Ihrem Workflow eine Testaktion hinzuzufügen.

### Um eine Testaktion hinzuzufügen

- Folgen Sie den Anweisungen in [Hinzufügen der Build-Aktion](#). Die Anweisungen zum Hinzufügen einer Build-Aktion und einer Testaktion sind exakt identisch.

## Definition der Testaktion

Die Testaktion ist als ein Satz von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Aktionsreferenz zum Erstellen und Testen](#) in der [Referenz zur Workflow-Definition](#).

## Berichte konfigurieren

In diesem Abschnitt wird beschrieben, wie Sie einen Qualitätsbericht konfigurieren.

### Themen

- [Konfiguration von Autodiscovery und manuellen Berichten](#)
- [Erfolgskriterien für Berichte konfigurieren](#)
- [Beispiel: Berichte konfigurieren](#)
- [SARIF-Unterstützung für Software Composition Analysis und Static Analysis Reports](#)

## Konfiguration von Autodiscovery und manuellen Berichten

Wenn die automatische Erkennung aktiviert ist, werden alle Eingaben, die an die Aktion übergeben wurden, und alle von der Aktion selbst generierten Dateien CodeCatalyst durchsucht. Dabei wird nach Berichten über Tests, Codeabdeckung, Software Composition Analysis (SCA) und statische Analyse (SA) gesucht. Sie können jeden dieser Berichte in CodeCatalyst anzeigen und bearbeiten.

Sie können auch manuell konfigurieren, welche Berichte generiert werden. Sie können den Typ des Berichts, den Sie generieren möchten, sowie das Dateiformat angeben. Weitere Informationen finden Sie unter [Typen von Testberichten](#).

## Erfolgskriterien für Berichte konfigurieren

Sie können die Werte festlegen, die die Erfolgskriterien für einen Test-, Codeabdeckungs-, Software Composition Analysis (SCA) oder Static Analysis (SA) -Bericht bestimmen.

Erfolgskriterien sind Schwellenwerte, die bestimmen, ob ein Bericht erfolgreich ist oder nicht. CodeCatalyst generiert zunächst Ihren Bericht, bei dem es sich um einen Test-, Codeabdeckungs-, SCA- oder SA-Bericht handeln kann, und wendet dann die Erfolgskriterien auf die generierten Berichte an. Anschließend wird angezeigt, ob und in welchem Umfang die Erfolgskriterien erfüllt wurden. Wenn ein Bericht die angegebenen Erfolgskriterien nicht erfüllt, schlägt die CodeCatalyst Aktion fehl, mit der die Erfolgskriterien angegeben wurden.



Wenn Sie beispielsweise die Erfolgskriterien für Ihren SCA-Bericht festlegen, lauten die gültigen Sicherheitsanfälligkeitwerte zwischen dem schwersten und dem geringsten CRITICAL Schweregrad: HIGHMEDIUM,,LOW,INFORMATIONAL. Wenn Sie die Kriterien für die Suche nach einer Sicherheitsanfälligkeit mit ihrem HIGH Schweregrad festlegen, schlägt der Bericht fehl, wenn entweder mindestens eine Sicherheitsanfälligkeit mit HIGH Schweregrad oder keine Sicherheitslücken mit HIGH Schweregrad vorhanden sind, aber mindestens eine Sicherheitsanfälligkeit mit einem höheren Schweregrad, z. B. eine Sicherheitsanfälligkeit mit CRITICAL Schweregrad.

Wenn Sie keine Erfolgskriterien angeben, gehen Sie wie folgt vor:

- Der CodeCatalyst Bericht, der auf der Grundlage Ihrer Rohberichte generiert wird, zeigt keine Erfolgskriterien an.
- Erfolgskriterien werden nicht verwendet, um zu bestimmen, ob die zugehörige Workflow-Aktion erfolgreich war oder nicht.

## Visual

Um Erfolgskriterien zu konfigurieren

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie einen Workflow aus, der eine Aktion enthält, die einen Bericht generiert. Dies ist der Bericht, für den Sie Erfolgskriterien anwenden möchten. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
3. Wählen Sie Bearbeiten aus.
4. Wählen Sie Visual.
5. Wählen Sie im Workflow-Diagramm die Aktion aus, die Sie für die Generierung von CodeCatalyst Berichten konfiguriert haben.
6. Wählen Sie die Registerkarte Outputs.
7. Wählen Sie unter Automatische Erkennung von Berichten oder unter Berichte manuell konfigurieren die Option Erfolgskriterien aus.

Erfolgskriterien werden angezeigt. Abhängig von Ihrer vorherigen Auswahl werden Ihnen möglicherweise einige oder alle dieser Optionen angezeigt:

Erfolgsquote

Geben Sie den Prozentsatz der Tests in einem Testbericht an, die bestanden werden müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Zu den gültigen Werten gehören Dezimalzahlen. Zum Beispiel: 50, 60.5. Die Kriterien für die Erfolgsquote werden nur auf Testberichte angewendet. Weitere Informationen zu Testberichten finden Sie unter [Testberichte](#).

### Abdeckung der Leitungen

Geben Sie den Prozentsatz der Zeilen in einem Bericht über die Codeabdeckung an, die abgedeckt sein müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Zu den gültigen Werten gehören Dezimalzahlen. Zum Beispiel: 50, 60.5. Die Kriterien für die Leitungsabdeckung werden nur auf Berichte zur Codeabdeckung angewendet. Weitere Informationen zu Berichten über die Codeabdeckung finden Sie unter [Berichte zur Codeabdeckung](#).

### Abdeckung der Filialen

Geben Sie den Prozentsatz der Filialen in einem Bericht über die Codeabdeckung an, die abgedeckt sein müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Gültige Werte beinhalten Dezimalzahlen. Zum Beispiel: 50, 60.5. Die Kriterien für die Abdeckung von Filialen werden nur auf Berichte zur Codeabdeckung angewendet. Weitere Informationen zu Berichten über die Codeabdeckung finden Sie unter [Berichte zur Codeabdeckung](#).

### Sicherheitslücken (SCA)

Geben Sie die maximale Anzahl und den Schweregrad der Sicherheitslücken an, die im SCA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert werden kann. Um Sicherheitslücken zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Sicherheitsanfälligkeiten, die Sie in die Zählung einbeziehen möchten. Gültige Werte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählen HIGHHIGH, werden alle CRITICAL Sicherheitslücken gezählt.

- Die maximale Anzahl von Sicherheitslücken mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Kriterien für Sicherheitslücken werden nur auf SCA-Berichte angewendet.

Weitere Informationen zu SCA-Berichten finden Sie unter [Berichte zur Analyse der Softwarezusammensetzung](#).

## Bugs

Geben Sie die maximale Anzahl und den Schweregrad der Fehler an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Um Fehler zu spezifizieren, müssen Sie Folgendes angeben:

- Der minimale Schweregrad der Fehler, die Sie in die Zählung einbeziehen möchten. Gültige Werte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählenHIGH, dann werden HIGH alle CRITICAL Bugs gezählt.

- Die maximale Anzahl von Fehlern mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Bug-Kriterien werden nur auf PyLint ESLint SA-Berichte angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

## Sicherheitslücken

Geben Sie die maximale Anzahl und den Schweregrad der Sicherheitslücken an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Um Sicherheitslücken zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Sicherheitslücken, die Sie in die Zählung einbeziehen möchten. Gültige Werte (vom höchsten bis zum geringsten Schweregrad) sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählenHIGH, dann werden HIGH werden CRITICAL Sicherheitslücken gezählt.

- Die maximale Anzahl von Sicherheitslücken mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Kriterien für Sicherheitslücken werden nur auf PyLint Berichte von ESLint SA angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

### Probleme mit der Qualität

Geben Sie die maximale Anzahl und den Schweregrad der Qualitätsprobleme an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert werden kann. Um Qualitätsprobleme zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Qualitätsprobleme, die Sie in die Zählung einbeziehen möchten. Gültige Werte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie sich beispielsweise dafür entscheiden HIGHHIGH, werden CRITICAL Qualitätsprobleme gezählt.

- Die maximale Anzahl von Qualitätsproblemen mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Kriterien für Qualitätsprobleme werden nur auf PyLint Berichte von ESLint SA angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

8. Wählen Sie Commit (Übergeben).
9. Führen Sie Ihren Workflow so aus, dass Erfolgskriterien auf Ihre Rohberichte CodeCatalyst angewendet werden, und generieren Sie die zugehörigen CodeCatalyst Berichte erneut, einschließlich der Informationen zu den Erfolgskriterien. Weitere Informationen finden Sie unter [Einen Workflow-Lauf starten](#).

## YAML

Um Erfolgskriterien zu konfigurieren

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie einen Workflow aus, der eine Aktion enthält, die einen Bericht generiert. Dies ist der Bericht, für den Sie Erfolgskriterien anwenden möchten. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
3. Wählen Sie Bearbeiten aus.
4. Wählen Sie YAML.
5. Wählen Sie im Workflow-Diagramm die Aktion aus, die Sie für die Generierung von CodeCatalyst Berichten konfiguriert haben.
6. Wählen Sie im Detailbereich die Registerkarte Ausgaben aus.
7. Fügen Sie in der Aktion, im `AutoDiscoverReports` Abschnitt oder im `Reports` Abschnitt eine `SuccessCriteriaEigenschaft` zusammen mit `PassRate`, `LineCoverage`, `BranchCoverage`, `Vulnerabilities`, `StaticAnalysisBug`, `StaticAnalysisSecurity`, und `StaticAnalysisQuality` Eigenschaften hinzu.

Eine Erläuterung der einzelnen Eigenschaften finden Sie im [Aktionsreferenz zum Erstellen und Testen](#).

8. Wählen Sie Commit (Übergeben).
9. Führen Sie Ihren Workflow so aus, dass Erfolgskriterien auf Ihre Rohberichte CodeCatalyst angewendet werden, und generieren Sie die zugehörigen CodeCatalyst Berichte erneut, einschließlich der Informationen zu den Erfolgskriterien. Weitere Informationen zum Starten eines Workflows finden Sie unter [Einen Workflow-Lauf starten](#).

## Beispiel: Berichte konfigurieren

Das folgende Beispiel zeigt, wie Sie vier Berichte manuell konfigurieren: einen Testbericht, einen Bericht zur Codeabdeckung, einen Analysebericht zur Softwarezusammensetzung und einen statischen Analysebericht.

```
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
MySCAReport:
  Format: SARIFSCA
  IncludePaths:
    - output/sca/reports.xml
  SuccessCriteria:
    Vulnerabilities:
      Number: 5
      Severity: HIGH
MySAReport:
  Format: ESLINTJSON
  IncludePaths:
    - output/static/eslint.xml
  SuccessCriteria:
    StaticAnalysisBug:
      Number: 10
      Severity: MEDIUM
    StaticAnalysisSecurity:
      Number: 5
      Severity: CRITICAL
    StaticAnalysisQuality:
      Number: 0
      Severity: INFORMATIONAL
```

## SARIF-Unterstützung für Software Composition Analysis und Static Analysis Reports

SARIF (Static Analysis Results Interchange Format) ist ein Ausgabedateiformat, das in den Berichten Software Composition Analysis und Static Analysis unter verfügbar ist. CodeCatalyst Das folgende Beispiel zeigt, wie SARIF in einem statischen Analysebericht manuell konfiguriert wird:

```

Reports:
  MySARReport:
    Format: SARIFSA
    IncludePaths:
      - output/sa_report.json
    SuccessCriteria:
      StaticAnalysisFinding:
        Number: 25
        Severity: HIGH

```

CodeCatalyst unterstützt die folgenden SARIF-Eigenschaften, mit denen Sie die Darstellung der Analyseergebnisse in Ihren Berichten optimieren können.

## Themen

- [sarifLog-Objekt](#)
- [run-Objekt](#)
- [toolComponent-Objekt](#)
- [reportingDescriptor-Objekt](#)
- [result-Objekt](#)
- [location-Objekt](#)
- [physicalLocation-Objekt](#)
- [logicalLocation-Objekt](#)
- [fix-Objekt](#)

## sarifLog-Objekt

Name	Erforderlich	Beschreibung
\$schema	Ja	<a href="#">Die URI des SARIF-JSON-Schemas für Version 2.1.0.</a>
version	Ja	CodeCatalyst unterstützt nur SARIF Version 2.1.0.
runs[]	Ja	Eine SARIF-Datei enthält ein Array von einem oder

Name	Erforderlich	Beschreibung
		mehreren Durchläufen, von denen jeder einen einzelnen Lauf des Analysetools darstellt

### run-Objekt

Name	Erforderlich	Beschreibung
<code>tool.driver</code>	Ja	Ein <code>toolComponent</code> Objekt, das das Analysetool beschreibt.
<code>tool.name</code>	Nein	Eine Eigenschaft, die den Namen des Tools angibt, das zur Durchführung der Analyse verwendet wird.
<code>results[]</code>	Ja	Die Ergebnisse des Analysetools, die am angezeigt werden CodeCatalyst.

### toolComponent-Objekt

Name	Erforderlich	Beschreibung
<code>name</code>	Ja	Der Name des Analysetools.
<code>properties.artifactScanned</code>	Nein	Die Gesamtzahl der vom Tool analysierten Artefakte.
<code>rules[]</code>	Ja	Eine Reihe von <code>reportingDescriptor</code> Objekten, die Regeln darstellen. Auf der Grundlage dieser Regeln



Name	Erforderlich	Beschreibung
		findet das Analysetool Probleme im analysierten Code.

### **reportingDescriptor**-Objekt

Name	Erforderlich	Beschreibung
<code>id</code>	Ja	Die eindeutige Kennung für die Regel, die verwendet wird, um auf ein Ergebnis zu verweisen.  Maximale Länge: 1.024 Zeichen
<code>name</code>	Nein	Der Anzeigename der Regel.  Maximale Länge: 1.024 Zeichen
<code>shortDescription.text</code>	Nein	Eine verkürzte Beschreibung der Regel.  Maximale Länge: 3.000 Zeichen
<code>fullDescription.text</code>	Nein	Eine vollständige Beschreibung der Regel.  Maximale Länge: 3.000 Zeichen
<code>helpUri</code>	Nein	Eine Zeichenfolge, die so lokalisiert werden kann, dass sie den absoluten URI der

Name	Erforderlich	Beschreibung
		<p>Primärdokumentation für die Regel enthält.</p> <p>Maximale Länge: 3.000 Zeichen</p>
properties.unscore	Nein	Eine Markierung, die angibt, ob das Scan-Ergebnis bewertet wurde.
properties.score.severity	Nein	<p>Ein fester Satz von Zeichenketten, die den Schweregrad des Ergebnisses angeben.</p> <p>Maximale Länge: 1.024 Zeichen</p>
properties.cvssv3_baseSeverity	Nein	Eine qualitative Bewertung des Schweregrads des <a href="#">Common Vulnerability Scoring System v3.1</a> .
properties.cvssv3_baseScore	Nein	Ein CVSS v3-Basiswert im Bereich von <a href="#">0,0 bis 10,0</a> .
properties.cvssv2_severity	Nein	Wenn CVSS v3-Werte nicht verfügbar sind, wird nach CVSS v2-Werten CodeCatalyst gesucht.
properties.cvssv2_score	Nein	Ein CVSS v2-Basiswert im Bereich von <a href="#">0,0 bis 10,0</a> .

Name	Erforderlich	Beschreibung
<code>properties.severity</code>	Nein	Ein fester Satz von Zeichenketten, die den Schweregrad des Ergebnisses angeben.  Maximale Länge: 1.024 Zeichen
<code>defaultConfiguration.level</code>	Nein	Der Standardschweregrad einer Regel.

### result-Objekt

Name	Erforderlich	Beschreibung
<code>ruleId</code>	Ja	Die eindeutige Kennung für die Regel, die verwendet wird, um auf ein Ergebnis zu verweisen.  Maximale Länge: 1.024 Zeichen
<code>ruleIndex</code>	Ja	Der Index der zugehörigen Regel in der Tool-Komponente <code>rules[]</code> .
<code>message.text</code>	Ja	Eine Meldung, die das Ergebnis beschreibt und die Meldung für jedes Ergebnis anzeigt.  Maximale Länge: 3.000 Zeichen
<code>rank</code>	Nein	Ein Wert zwischen 0,0 und einschließlich 100,0, der die

Name	Erforderlich	Beschreibung
		Priorität oder Wichtigkeit des Ergebnisses angibt. Auf dieser Skala steht der Wert 0,0 für die niedrigste Priorität und 100,0 für die höchste Priorität.
level	Nein	Der Schweregrad des Ergebnisses.  Maximale Länge: 1.024 Zeichen
properties.unscore	Nein	Eine Markierung, die angibt, ob das Scanergebnis bewertet wurde.
properties.score.severity	Nein	Ein fester Satz von Zeichenketten, die den Schweregrad des Ergebnisses angeben.  Maximale Länge: 1.024 Zeichen
properties.cvssv3_baseSeverity	Nein	Eine qualitative Bewertung des Schweregrads des <a href="#">Common Vulnerability Scoring System v3.1</a> .
properties.cvssv3_baseScore	Nein	Ein CVSS v3-Basiswert im Bereich von <a href="#">0,0 bis 10,0</a> .
properties.cvssv2_severity	Nein	Wenn CVSS v3-Werte nicht verfügbar sind, wird nach CVSS v2-Werten CodeCatalyst gesucht.

Name	Erforderlich	Beschreibung
<code>properties.cvssv2_score</code>	Nein	Ein CVSS v2-Basiswert im Bereich von <a href="#">0,0 bis 10,0</a> .
<code>properties.severity</code>	Nein	Ein fester Satz von Zeichenketten, die den Schweregrad des Ergebnisses angeben.  Maximale Länge: 1.024 Zeichen
<code>locations[]</code>	Ja	Die Gruppe von Orten, an denen das Ergebnis erkannt wurde. Es sollte nur ein Standort angegeben werden, es sei denn, das Problem kann nur behoben werden, indem an jeder angegebenen Position eine Änderung vorgenommen wird. CodeCatalyst verwendet den ersten Wert in der Ortsmatrix, um das Ergebnis zu kommentieren.  Maximale Anzahl von <code>location</code> Objekten: 10
<code>relatedLocations[]</code>	Nein	Eine Liste zusätzlicher Ortsverweise im Ergebnis.  Maximale Anzahl von <code>location</code> Objekten: 50

Name	Erforderlich	Beschreibung
<code>fixes[]</code>	Nein	Eine Reihe von <code>fix</code> Objekten, die der Empfehlung des Scan-Tools entsprechen. CodeCatalyst verwendet die erste Empfehlung im <code>fixes</code> Array.

### **location**-Objekt

Name	Erforderlich	Beschreibung
<code>physicalLocation</code>	Ja	Identifiziert das Artefakt und die Region.
<code>logicalLocations[]</code>	Nein	Die Gruppe von Orten, die namentlich ohne Bezugnahme auf das Artefakt beschrieben werden.

### **physicalLocation**-Objekt

Name	Erforderlich	Beschreibung
<code>artifactLocation.uri</code>	Ja	Der URI, der den Speicherort eines Artefakts angibt, normalerweise eine Datei, die entweder im Repository oder während eines Builds generiert wurde.
<code>fileLocation.uri</code>	Nein	Der Fallback-URI, der den Speicherort der Datei angibt. Dieser Wert wird verwendet, wenn er leer <code>artifactL</code>

Name	Erforderlich	Beschreibung
		<code>location.uri</code> zurückgegeben wird.
<code>region.startLine</code>	Ja	Die Zeilennummer des ersten Zeichens in der Region.
<code>region.startColumn</code>	Ja	Die Spaltennummer des ersten Zeichens in der Region.
<code>region.endLine</code>	Ja	Die Zeilennummer des letzten Zeichens in der Region.
<code>region.endColumn</code>	Ja	Die Spaltennummer des letzten Zeichens in der Region.

### **logicalLocation**-Objekt

Name	Erforderlich	Beschreibung
<code>fullyQualifiedname</code>	Nein	Zusätzliche Informationen, die den Speicherort des Ergebnisses beschreiben.  Maximale Länge: 1.024 Zeichen

### **fix**-Objekt

Name	Erforderlich	Beschreibung
<code>description.text</code>	Nein	Eine Meldung, die für jedes Ergebnis eine Empfehlung enthält.

Name	Erforderlich	Beschreibung
		Maximale Länge: 3.000 Zeichen
<code>artifactChanges.[0].artifactLocation.uri</code>	Nein	Die URI, die den Speicherort des Artefakts angibt, das aktualisiert werden muss.

## Benutzen universal-test-runner

Testaktionen sind in das Open-Source-Befehlszeilentool `universal-test-runner` integriert. Dieses Tool bietet erweiterte Testfunktionen, z. B. die Wiederholung eines oder mehrerer Testfälle aus einem Testbericht. `universal-test-runner` verwendet das [Test Execution Protocol](#), um Ihre Tests für eine beliebige Sprache in einem bestimmten Framework auszuführen. `universal-test-runner` unterstützt die folgenden Frameworks:

- [Gradle](#)
- [Scherz](#)
- [Seherin](#)
- [Pytest](#)
- [.NET](#)

`universal-test-runner` ist nur auf den kuratierten Bildern für Testaktionen installiert. Wenn Sie eine Testaktion für die Verwendung eines benutzerdefinierten Docker Hubs oder Amazon ECR konfigurieren, müssen Sie sie manuell installieren, um erweiterte Testfunktionen `universal-test-runner` zu aktivieren. Installieren Sie dazu Node.js (14 oder höher) auf dem Image und installieren Sie es anschließend `universal-test-runner` mithilfe des Shell-Befehls. - Run: `npm install -g @aws/universal-test-runner` Weitere Informationen zur Installation von Node.js in Ihrem Container mithilfe von Shell-Befehlen finden Sie unter [Node Version Manager installieren und aktualisieren](#).

Weitere Informationen zu `universal-test-runner` finden Sie unter [Was ist universal-test-runner?](#)



## Visual

Zur Verwendung universal-test-runner im Visual Editor

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie den Namen Ihres Workflows.
4. Wählen Sie Bearbeiten aus.
5. Wählen Sie Visual.
6. Wählen Sie Aktionen.
7. Wählen Sie unter Aktionen die Option Test aus.
8. Füllen Sie auf der Registerkarte Konfiguration das Feld Shell-Befehle aus, indem Sie den Beispielcode mit den unterstützten Frameworks Ihrer Wahl aktualisieren. Um beispielsweise ein unterstütztes Framework zu verwenden, würden Sie einen Run Befehl verwenden, der dem folgenden ähnelt.

```
- Run: run-tests <framework>
```

Wenn das von Ihnen gewünschte Framework nicht unterstützt wird, sollten Sie einen benutzerdefinierten Adapter oder Runner beisteuern. Eine Beschreibung des Felds Shell-Befehle finden Sie unter [Steps](#).

9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Zur Verwendung universal-test-runner im YAML-Editor

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie den Namen Ihres Workflows.
4. Wählen Sie Bearbeiten aus.

5. Wählen Sie YAML.
6. Wählen Sie Aktionen.
7. Wählen Sie unter Aktionen die Option Test aus.
8. Ändern Sie den YAML-Code nach Ihren Bedürfnissen. Um beispielsweise ein unterstütztes Framework zu verwenden, würden Sie einen Run Befehl verwenden, der dem folgenden ähnelt.

```
Configuration:  
  Steps:  
    - Run: run-tests <framework>
```

Wenn das von Ihnen gewünschte Framework nicht unterstützt wird, sollten Sie einen benutzerdefinierten Adapter oder Runner beisteuern. Eine Beschreibung der Eigenschaft Steps finden Sie unter [Steps](#).

9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Bewährte Methoden

Wenn Sie die von bereitgestellten Testfunktionen verwenden CodeCatalyst, empfehlen wir Ihnen, diese bewährten Methoden zu befolgen.

### Themen

- [Automatische Erkennung](#)
- [Erfolgskriterien](#)
- [Pfade einschließen/ausschließen](#)

## Automatische Erkennung

Bei der Konfiguration von Aktionen in CodeCatalyst können Sie mit der automatischen Erkennung automatisch die Ausgaben verschiedener Tools, wie z. B. JUnit-Testberichte, ermitteln und daraus relevante CodeCatalyst Berichte generieren. Durch die automatische Erkennung wird sichergestellt, dass weiterhin Berichte generiert werden, auch wenn sich die Namen oder Pfade zu den erkannten

Ergebnissen ändern. Wenn neue Dateien hinzugefügt werden, werden diese CodeCatalyst automatisch erkannt und relevante Berichte erstellt. Wenn Sie jedoch die automatische Erkennung verwenden, ist es wichtig, einige der folgenden Aspekte dieser Funktion zu berücksichtigen:

- Wenn Sie in Ihrer Aktion die automatische Erkennung aktivieren, weisen alle automatisch erkannten Berichte desselben Typs dieselben Erfolgskriterien auf. Beispielsweise würden gemeinsame Kriterien wie die Mindestbestehensquote für alle automatisch erkannten Testberichte gelten. Wenn Sie unterschiedliche Kriterien für Berichte desselben Typs benötigen, müssen Sie jeden dieser Berichte explizit konfigurieren.
- Mit der automatischen Erkennung können auch Berichte gefunden werden, die von Ihren Abhängigkeiten erstellt wurden. Wenn Erfolgskriterien konfiguriert sind, schlägt die Aktion für diese Berichte möglicherweise fehl. Dieses Problem kann durch eine Aktualisierung der Konfiguration für den Ausschlusspfad behoben werden.
- Es kann nicht garantiert werden, dass bei der automatischen Erkennung jedes Mal dieselbe Berichtsliste erstellt wird, da die Aktion zur Laufzeit gescannt wird. Wenn Sie möchten, dass ein bestimmter Bericht immer erstellt wird, sollten Sie Berichte explizit konfigurieren. Wenn beispielsweise Tests im Rahmen Ihres Builds nicht mehr ausgeführt werden, würde das Testframework keine Ausgaben erzeugen und folglich würde kein Testbericht erstellt werden, und die Aktion könnte erfolgreich sein. Wenn Sie möchten, dass der Erfolg Ihrer Aktion von diesem speziellen Test abhängt, müssen Sie diesen Bericht explizit konfigurieren.

#### Tip

Wenn Sie mit einem neuen oder vorhandenen Projekt beginnen, verwenden Sie die automatische Erkennung für das gesamte Projektverzeichnis (einschließlich `**/*`). Dadurch wird die Berichtsgenerierung für alle Dateien in Ihrem Projekt aufgerufen, einschließlich der Dateien in Unterverzeichnissen.

Weitere Informationen finden Sie unter [Berichte konfigurieren](#).

## Erfolgskriterien

Sie können Qualitätsgrenzwerte für Ihre Berichte durchsetzen, indem Sie Erfolgskriterien konfigurieren. Wenn beispielsweise zwei Codeabdeckungsberichte automatisch erkannt wurden, einer mit einer Leitungsabdeckung von 80% und der andere mit einer Leitungsabdeckung von 60%, haben Sie die folgenden Optionen:

- Legen Sie die Erfolgskriterien für die automatische Erkennung für die Leitungsabdeckung auf 80% fest. Dies würde dazu führen, dass der erste Bericht erfolgreich und der zweite Bericht fehlschlägt, was dazu führen würde, dass die gesamte Aktion fehlschlägt. Um den Workflow zu entsperren, fügen Sie Ihrem Projekt neue Tests hinzu, bis die Zeilenabdeckung für den zweiten Bericht 80% überschreitet.
- Legen Sie die Erfolgskriterien für die automatische Erkennung für die Leitungsabdeckung auf 60% fest. Dies würde dazu führen, dass beide Berichte erfolgreich sind, was dazu führen würde, dass die Aktion erfolgreich wäre. Sie könnten dann daran arbeiten, die Codeabdeckung im zweiten Bericht zu erhöhen. Mit diesem Ansatz können Sie jedoch nicht garantieren, dass der Erfassungsgrad im ersten Bericht nicht unter 80% fällt.
- Konfigurieren Sie einen oder beide Berichte explizit, indem Sie den visuellen Editor verwenden oder für jeden Bericht einen expliziten YAML-Abschnitt und -Pfad hinzufügen. Auf diese Weise könnten Sie separate Erfolgskriterien und benutzerdefinierte Namen für jeden Bericht konfigurieren. Bei diesem Ansatz könnte die Aktion jedoch fehlschlagen, wenn sich die Berichtspfade ändern.

Weitere Informationen finden Sie unter [Erfolgskriterien für Berichte konfigurieren](#).

## Pfade einschließen/ausschließen

Bei der Überprüfung der Aktionsergebnisse können Sie die Liste der Berichte, die generiert werden, anpassen, CodeCatalyst indem Sie und konfigurieren `IncludePaths`. `ExcludePaths`

- Geben `IncludePaths` Sie hier die Dateien und Dateipfade CodeCatalyst an, die Sie bei der Suche nach Berichten berücksichtigen möchten. Wenn Sie beispielsweise angeben, dass das gesamte Build-Image `"/test/report/*"`, das von der Aktion verwendet wurde, nach dem `/test/report/` Verzeichnis CodeCatalyst durchsucht wird, durchsucht wird. Wenn das Verzeichnis gefunden CodeCatalyst wird, wird in diesem Verzeichnis nach Berichten gesucht.

### Note

Bei manuell konfigurierten Berichten `IncludePaths` muss es sich um ein Glob-Muster handeln, das einer einzelnen Datei entspricht.

- Geben `ExcludePaths` Sie hier die Dateien und Dateipfade an, die Sie bei der Suche nach Berichten ausschließen möchten CodeCatalyst . Wenn Sie beispielsweise angeben `"/test/reports/**/*"`, CodeCatalyst wird nicht nach Dateien im `/test/reports/` Verzeichnis

gesucht. Um alle Dateien in einem Verzeichnis zu ignorieren, verwenden Sie das `**/*` Glob-Muster.

Im Folgenden finden Sie Beispiele für mögliche Glob-Muster.

Muster	Beschreibung
<code>*.*</code>	Entspricht allen Objektnamen im aktuellen Verzeichnis, die einen Punkt enthalten
<code>*.xml</code>	Entspricht allen Objektnamen im aktuellen Verzeichnis, die mit enden <code>.xml</code>
<code>*.{xml,txt}</code>	Entspricht allen Objektnamen im aktuellen Verzeichnis, die mit <code>.xml</code> oder enden <code>.txt</code>
<code>**/*.xml</code>	Entspricht Objektnamen in allen Verzeichnissen, die mit enden <code>.xml</code>
<code>testFolder</code>	Entspricht einem <code>testFolder</code> aufgerufenen Objekt und behandelt es als Datei
<code>testFolder/*</code>	Entspricht Objekten auf einer Ebene des Unterordners <code>testFolder</code> , z. B. <code>testFolder/file.xml</code>
<code>testFolder/**</code>	Entspricht Objekten auf zwei Ebenen des Unterordners <code>testFolder</code> , z. B. <code>testFolder/reportsFolder/file.xml</code>
<code>testFolder/**</code>	Entspricht dem Unterordner <code>testFolder</code> und Dateien unter <code>testFolder</code> , wie z. B. <code>testFolder/file.xml</code> und <code>testFolder/otherFolder/file.xml</code>

CodeCatalyst interpretiert die Glob-Muster wie folgt:

- Der Schrägstrich (/) trennt Verzeichnisse in Dateipfaden.
- Das Sternchen (\*) entspricht einem oder mehreren Zeichen einer Namenskomponente ohne Überschreiten der Ordnergrenzen.
- Ein doppeltes Sternchen (\*\*) steht für null oder mehr Zeichen einer Namenskomponente in allen Verzeichnissen.

#### Note

ExcludePathshat Vorrang vor. IncludePaths Wenn beide IncludePaths den gleichen Ordner ExcludePaths enthalten, wird dieser Ordner nicht nach Berichten durchsucht.

## Mit Tests arbeiten

In diesem Abschnitt wird beschrieben, wie Sie Ihre Testaktionen anzeigen und konfigurieren.

### Themen

- [Ergebnisse einer Testaktion anzeigen](#)
- [Fehlgeschlagene Tests überspringen](#)
- [Testfälle erneut versuchen](#)

## Ergebnisse einer Testaktion anzeigen

Verwenden Sie die folgenden Anweisungen, um die Ergebnisse einer Testaktion, einschließlich der generierten Protokolle, Berichte und Variablen, anzuzeigen.

So zeigen Sie die Ergebnisse einer Testaktion an

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
3. Wählen Sie im Workflow-Diagramm den Namen Ihrer Testaktion aus (z. B. Test).
4. Um die von einer Aktion generierten Protokolle anzuzeigen, wählen Sie Logs aus. Die Protokolle für die verschiedenen Aktionsphasen werden angezeigt. Sie können die Protokolle nach Bedarf erweitern oder reduzieren.

5. Um die durch die Testaktion erstellten Testberichte anzuzeigen, wählen Sie Berichte aus. Weitere Informationen finden Sie unter [Typen von Testberichten](#).
6. Um die für die Testaktion verwendete Konfiguration anzuzeigen, wählen Sie Konfiguration. Weitere Informationen finden Sie unter [Testaktion hinzufügen](#).
7. Um die von der Testaktion verwendeten Variablen anzuzeigen, wählen Sie Variablen. Weitere Informationen finden Sie unter [Arbeiten mit Variablen](#).

## Fehlgeschlagene Tests überspringen

Wenn Ihre Aktion mehr als einen Testbefehl umfasst, möchten Sie möglicherweise zulassen, dass nachfolgende Testbefehle in der Aktion ausgeführt werden, auch wenn ein vorheriger Befehl fehlschlägt. In den folgenden Befehlen möchten Sie beispielsweise möglicherweise immer `ausführentest2`, auch wenn sie `test1` fehlschlagen.

### Steps:

- Run: `npm install`
- Run: `npm run test1`
- Run: `npm run test2`

Wenn ein Schritt einen Fehler zurückgibt, CodeCatalyst stoppt Amazon normalerweise die Workflow-Aktion und markiert sie als fehlgeschlagen. Sie können zulassen, dass die Aktionsschritte weiterhin ausgeführt werden, indem Sie die Fehlerausgabe an umleiten. `null` Sie können dies tun, indem Sie dem Befehl etwas `2>/dev/null` hinzufügen. Mit dieser Änderung würde das vorherige Beispiel wie folgt aussehen.

### Steps:

- Run: `npm install`
- Run: `npm run test1 2>/dev/null`
- Run: `npm run test2`

Im zweiten Codeausschnitt wird der Status des `npm install` Befehls berücksichtigt, aber alle vom `npm run test1` Befehl zurückgegebenen Fehler werden ignoriert. Infolgedessen wird der `npm run test2` Befehl ausgeführt. Auf diese Weise können Sie beide Berichte gleichzeitig anzeigen, unabhängig davon, ob ein Fehler auftritt.

## Testfälle erneut versuchen

Wenn Ihr Bericht aufgrund mehrerer Testfälle fehlschlägt, können Sie nur diese einzelnen Tests wiederholen. Auf diese Weise können Sie schnell die Qualität Ihrer Testfälle überprüfen und die nächsten Schritte zur Lösung Ihrer Probleme festlegen, z. B. die Aktivierung einer unterbrochenen Abhängigkeit oder die Initiierung einer Workflow-Wiederholung. Ihre Testaktion beinhaltet die Option `universal-test-runner`, nur ausgewählte Testfälle und nicht die gesamte Aktion erneut zu versuchen. Sie können jeweils nur einen Satz ausgewählter Testfälle pro Aktion und nur fünfmal pro Testbericht wiederholen. Weitere Informationen finden Sie unter [Benutzen universal-test-runner](#).

### Note

Wenn Sie Testfälle in einem Bericht erneut versuchen, hat dies keine Auswirkungen auf den Status des Workflows, der den ursprünglichen Bericht generiert hat.

Verwenden Sie die folgenden Anweisungen, um die Testfälle in Ihren Berichten erneut zu versuchen.

Um die Testfälle eines Berichts erneut zu versuchen

1. Klicken Sie im Navigationsbereich auf Berichte.
2. Wählen Sie den Namen Ihres Berichts. Sie können nach dem Namen, dem Status, dem Repository, der Branche oder dem Berichtstyp filtern.
3. Wählen Sie unter dem Namen des Berichts die Option Ergebnisse aus.
4. Wählen Sie die Testfälle aus, die Sie erneut versuchen möchten, wählen Sie Erneut ausführen und anschließend Ausgewählte Testfälle aus.
5. Wenn Ihr Wiederholungsversuch abgeschlossen ist, wählen Sie im Banner die Option Aktualisieren aus und sehen Sie sich die aktualisierten Ergebnisse an.

## Bereitstellung mithilfe von Workflows in CodeCatalyst

Mithilfe von [CodeCatalyst Workflows](#) können Sie Anwendungen und andere Ressourcen für verschiedene Ziele wie Amazon ECS und mehr bereitstellen. AWS Lambda

Themen

- [Wie stelle ich eine Anwendung bereit?](#)



- [Liste der Bereitstellungsaktionen](#)
- [Vorteile von Bereitstellungsaktionen](#)
- [Alternativen zur Bereitstellung von Aktionen](#)
- [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#)
- [Tutorial: Bereitstellen einer Anwendung in Amazon ECS](#)
- [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#)
- [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#)
- [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#)
- [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#)
- [Aktion „AWS CDK Deploy“ hinzufügen](#)
- [Arbeiten mit Bereitstellungen](#)

## Wie stelle ich eine Anwendung bereit?

Um eine Anwendung oder Ressource bereitzustellen CodeCatalyst, erstellen Sie zunächst einen Workflow und geben dann darin eine Bereitstellungsaktion an. Eine Bereitstellungsaktion ist ein Workflow-Baustein, der definiert, was Sie bereitstellen möchten, wo Sie es bereitstellen möchten und wie Sie es bereitstellen möchten (z. B. mit einem blauen/grünen Schema). Sie fügen Ihrem Workflow mithilfe des visuellen Editors oder des YAML-Editors der CodeCatalyst Konsole eine Bereitstellungsaktion hinzu.

Die allgemeinen Schritte zur Bereitstellung einer Anwendung oder Ressource lauten wie folgt.

So stellen Sie eine Anwendung bereit (Aufgaben auf hoher Ebene)

1. In Ihrem CodeCatalyst Projekt fügen Sie Quellcode für eine Anwendung hinzu, die Sie bereitstellen möchten. Weitere Informationen finden Sie unter [Arbeiten mit Quell-Repositorys in CodeCatalyst](#).
2. In Ihrem CodeCatalyst Projekt erstellen Sie einen Workflow. In diesem Workflow definieren Sie, wie Ihre Anwendung erstellt, getestet und bereitgestellt werden soll. Weitere Informationen finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#).
3. Im Workflow fügen Sie einen Auslöser, eine Build-Aktion und optional eine Testaktion hinzu. Weitere Informationen finden Sie unter [Arbeiten mit Auslösern](#), [Hinzufügen der Build-Aktion](#) und [Testaktion hinzufügen](#).

4. Im Workflow fügen Sie eine Bereitstellungsaktion hinzu. Sie können aus mehreren CodeCatalyst bereitgestellten Bereitstellungsaktionen für Ihre Anwendung für verschiedene Ziele wählen, z. B. Amazon ECS. (Sie können auch eine Build-Aktion oder eine GitHub Aktion verwenden, um Ihre Anwendung bereitzustellen. Weitere Informationen zur Build-Aktion und zu GitHub Aktionen finden Sie unter [Alternativen zur Bereitstellung von Aktionen](#).)
5. Sie starten den Workflow entweder manuell oder automatisch über einen Trigger. Der Workflow führt die Build-, Test- und Bereitstellungsaktionen nacheinander aus, um Ihre Anwendung und Ressourcen auf dem Ziel bereitzustellen. Weitere Informationen finden Sie unter [Einen Workflow-Lauf starten](#).

## Liste der Bereitstellungsaktionen

Die folgenden Bereitstellungsaktionen sind verfügbar:

- **AWS CloudFormation Stack bereitstellen** — Diese Aktion erstellt einen CloudFormation Stack auf der AWS Grundlage einer von Ihnen bereitgestellten [AWS CloudFormationAWS Serverless Application Model Vorlage oder Vorlage](#). Weitere Informationen finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#).
- **In Amazon ECS bereitstellen** — Diese Aktion registriert eine von Ihnen bereitgestellte [Aufgabendefinitionsdatei](#). Weitere Informationen finden Sie unter [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#).
- **Auf Kubernetes-Cluster bereitstellen** — Diese Aktion stellt eine Anwendung in einem Amazon Elastic Kubernetes Service Service-Cluster bereit. Weitere Informationen finden Sie unter [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#).
- **AWS CDK bereitstellen** — Diese Aktion stellt eine App bereit in [AWS CDK](#) AWS Weitere Informationen finden Sie unter [Aktion „AWS CDK Deploy“ hinzufügen](#).

### Note

Es gibt andere CodeCatalyst Aktionen, mit denen Ressourcen bereitgestellt werden können. Sie gelten jedoch nicht als Bereitstellungsaktionen, da ihre Bereitstellungsinformationen nicht auf der Seite Umgebungen angezeigt werden. Weitere Informationen zur Seite „Umgebungen“ und zum Anzeigen von Bereitstellungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Arbeiten mit Bereitstellungen](#).

## Vorteile von Bereitstellungsaktionen

Die Verwendung von Bereitstellungsaktionen innerhalb eines Workflows hat die folgenden Vorteile:

- **Bereitstellungsverlauf** — Sehen Sie sich einen Verlauf Ihrer Bereitstellungen an, um Änderungen an Ihrer bereitgestellten Software besser verwalten und kommunizieren zu können.
- **Rückverfolgbarkeit** — Verfolgen Sie den Status Ihrer Bereitstellungen über die CodeCatalyst Konsole und sehen Sie, wann und wo die einzelnen Anwendungsversionen bereitgestellt wurden.
- **Rollbacks** — Machen Sie Bereitstellungen automatisch rückgängig, wenn Fehler auftreten. Sie können auch Alarme konfigurieren, um Bereitstellungs-Rollbacks zu aktivieren.
- **Überwachung** — Beobachten Sie Ihre Implementierung, während sie die verschiedenen Phasen Ihres Workflows durchläuft.
- **Integration mit anderen CodeCatalyst Funktionen** — Speichern Sie den Quellcode und erstellen, testen und implementieren Sie ihn — alles von einer einzigen Anwendung aus.

## Alternativen zur Bereitstellung von Aktionen

Sie müssen keine Bereitstellungsaktionen verwenden, obwohl sie empfohlen werden, da sie die im vorherigen Abschnitt beschriebenen Vorteile bieten. Stattdessen können Sie die folgenden [CodeCatalyst Aktionen](#) verwenden:

- Eine Build-Aktion.

In der Regel verwenden Sie Build-Aktionen, wenn Sie eine Bereitstellung auf einem Ziel durchführen möchten, für das es keine entsprechende Bereitstellungsaktion gibt, oder wenn Sie mehr Kontrolle über das Bereitstellungsverfahren haben möchten. Weitere Informationen zur Verwendung von Build-Aktionen zur Bereitstellung von Ressourcen finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#).

- Eine GitHub Aktion.

Sie können eine [GitHub Aktion](#) innerhalb eines CodeCatalyst Workflows verwenden, um Anwendungen und Ressourcen bereitzustellen (anstelle einer CodeCatalyst Aktion). Informationen zur Verwendung von GitHub Aktionen innerhalb eines CodeCatalyst Workflows finden Sie unter [GitHub Aktionen hinzufügen](#).

Sie können auch die folgenden AWS Dienste verwenden, um Ihre Anwendung bereitzustellen, wenn Sie dafür keinen CodeCatalyst Workflow verwenden möchten:

- AWS CodeDeploy — siehe [Was ist CodeDeploy?](#)
- AWS CodeBuild und AWS CodePipeline — siehe [Was ist AWS CodeBuild?](#) und [Was ist AWS CodePipeline?](#)
- AWS CloudFormation — siehe [Was ist AWS CloudFormation?](#)

Verwenden Sie CodeDeploy CodeBuild, CodePipeline, und CloudFormation Services für komplexe Unternehmensbereitstellungen.

## Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation

In diesem Tutorial erfahren Sie, wie Sie mithilfe eines Workflows und eine serverlose Anwendung erstellen, testen und bereitstellen. AWS CloudFormation

Die Anwendung in diesem Tutorial ist eine einfache Webanwendung, die eine „Hello World“-Meldung ausgibt. Es besteht aus einer AWS Lambda Funktion und einem Amazon API Gateway, und Sie erstellen es mit der [AWS Serverless Application Model \(AWS SAM\)](#), einer Erweiterung von [AWS CloudFormation](#).

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie ein Quell-Repository](#)
- [Schritt 2: AWS Rollen erstellen](#)
- [Schritt 3: AWS Rollen hinzufügen zu CodeCatalyst](#)
- [Schritt 4: Erstellen Sie einen Amazon S3 S3-Bucket](#)
- [Schritt 5: Quelldateien hinzufügen](#)
- [Schritt 6: Erstellen Sie einen Workflow und führen Sie ihn aus](#)
- [Schritt 7: Nehmen Sie eine Änderung vor](#)
- [Bereinigen](#)

## Voraussetzungen

Bevor Sie beginnen:

- Sie benötigen einen CodeCatalyst Bereich mit einem verbundenen AWS Konto. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).
- In deinem Space benötigst du ein leeres CodeCatalyst Projekt mit dem Namen:

```
codecatalyst-cfn-project
```

Weitere Informationen finden Sie unter [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#).

- In Ihrem Projekt benötigen Sie eine CodeCatalyst Umgebung namens:

```
codecatalyst-cfn-environment
```

Konfigurieren Sie diese Umgebung wie folgt:

- Wählen Sie einen beliebigen Typ, z. B. „Keine Produktion“.
- Connect dein AWS Konto damit.

Weitere Informationen finden Sie unter [Arbeiten mit Umgebungen](#).

## Schritt 1: Erstellen Sie ein Quell-Repository

In diesem Schritt erstellen Sie ein Quell-Repository in CodeCatalyst. Dieses Repository wird verwendet, um die Quelldateien des Tutorials zu speichern, z. B. die Lambda-Funktionsdatei.

Weitere Hinweise zu Quell-Repositorys finden Sie unter [Erstellen eines Quell-Repositorys](#)

Um ein Quell-Repository zu erstellen

1. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus. CodeCatalyst
2. Wählen Sie Repository hinzufügen und anschließend Repository erstellen aus.
3. Geben Sie im Feld Repository-Name Folgendes ein:

```
codecatalyst-cfn-source-repository
```

4. Wählen Sie Erstellen.

Sie haben jetzt ein Repository mit dem Namen `codecatalyst-cfn-source-repository`.

## Schritt 2: AWS Rollen erstellen

In diesem Schritt erstellen Sie die folgenden AWS IAM-Rollen:

- **Rolle bereitstellen** — Erteilt der Aktion „AWS CloudFormation Stack CodeCatalyst bereitstellen“ die Berechtigung, auf Ihr AWS Konto und Ihren CloudFormation Dienst zuzugreifen, in dem Sie Ihre serverlose Anwendung bereitstellen werden. Die Aktion `AWS CloudFormation Stack bereitstellen` ist Teil Ihres Workflows.
- **Build-Rolle** — Erteilt der CodeCatalyst Build-Aktion die Berechtigung, auf Ihr AWS Konto zuzugreifen und in Amazon S3 zu schreiben, wo Ihr serverloses Anwendungspaket gespeichert wird. Die Build-Aktion ist Teil Ihres Workflows.
- **Stack-Rolle** — Erteilt die CloudFormation Berechtigung zum Lesen und Ändern der Ressourcen, die in der AWS SAM Vorlage angegeben sind, die Sie später bereitstellen werden. Erteilt auch die Erlaubnis für CloudWatch.

Weitere Informationen zu IAM-Rollen finden Sie unter [IAM-Rollen](#) im AWS Identity and Access Management Benutzerhandbuch.

### Note

Um Zeit zu sparen, können Sie anstelle der drei zuvor aufgeführten Rollen eine einzelne `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle, die so genannte Rolle, erstellen. Weitere Informationen finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfangreiche Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger wichtig ist. In diesem Tutorial wird davon ausgegangen, dass Sie die drei zuvor aufgeführten Rollen erstellen.

**Note**

Eine [Lambda-Ausführungsrolle](#) ist ebenfalls erforderlich, aber Sie müssen sie jetzt nicht erstellen, da die `sam-template.yml` Datei sie für Sie erstellt, wenn Sie den Workflow in Schritt 5 ausführen.

## Um eine Bereitstellungsrolle zu erstellen

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich an bei AWS.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
  - c. Wählen Sie im Navigationsbereich Policies aus.
  - d. Wählen Sie Richtlinie erstellen aus.
  - e. Wählen Sie den Tab JSON.
  - f. Löschen Sie den vorhandenen Code.
  - g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

}

**Note**

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie im Feld Name Folgendes ein:

```
codecatalyst-deploy-policy

```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Bereitstellungsrolle wie folgt:
  - a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
  - b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie.
  - c. Löschen Sie die bestehende benutzerdefinierte Vertrauensrichtlinie.
  - d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      }
    }
  ]
}

```



```
        },
        "Action": "sts:AssumeRole"
    }
]
}
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach dem entsprechenden Kontrollkästchen `codecatalyst-deploy-policy` und aktivieren Sie es.
- g. Wählen Sie Weiter aus.
- h. Geben Sie als Rollenname Folgendes ein:

**codecatalyst-deploy-role**

- i. Geben Sie als Rollenbeschreibung Folgendes ein:

**CodeCatalyst deploy role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Bereitstellungsrolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Rufen Sie den ARN für die Bereitstellungsrolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-deploy-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.

Die Übersichtsseite der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.

Sie haben jetzt die Bereitstellungsrolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

## Um eine Build-Rolle zu erstellen

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich an bei AWS.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
  - c. Wählen Sie im Navigationsbereich Policies aus.
  - d. Wählen Sie Richtlinie erstellen aus.
  - e. Wählen Sie den Tab JSON.
  - f. Löschen Sie den vorhandenen Code.
  - g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie im Feld Name Folgendes ein:

**codecatalyst-build-policy**

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Build-Rolle wie folgt:

- a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie.
- c. Löschen Sie die bestehende benutzerdefinierte Vertrauensrichtlinie.
- d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach dem entsprechenden Kontrollkästchen `codecatalyst-build-policy` und aktivieren Sie es.
- g. Wählen Sie Weiter aus.
- h. Geben Sie als Rollenname Folgendes ein:

**codecatalyst-build-role**

- i. Geben Sie als Rollenbeschreibung Folgendes ein:

**CodeCatalyst build role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Rufen Sie den ARN für die Build-Rolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-build-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.


Die Übersichtsseite der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.

Sie haben jetzt die Build-Rolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

### Um eine Stack-Rolle zu erstellen

1. Melden Sie sich AWS mit dem Konto an, in dem Sie Ihren Stack bereitstellen möchten.
2. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
3. Erstellen Sie die Stack-Rolle wie folgt:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Wählen Sie Rolle erstellen aus.
  - c. Wählen Sie einen AWS -Service aus.
  - d. Wählen Sie im Abschnitt Anwendungsfall eine Option CloudFormation aus der Drop-down-Liste aus.
  - e. Wählen Sie das Optionsfeld CloudFormation.
  - f. Wählen Sie unten Weiter aus.
  - g. Suchen Sie mithilfe des Suchfeldes nach den folgenden Berechtigungsrichtlinien und aktivieren Sie dann die entsprechenden Kontrollkästchen.

 Note

Wenn Sie nach einer Richtlinie suchen und sie nicht angezeigt wird, wählen Sie Filter löschen aus und versuchen Sie es erneut.

- CloudWatchFullAccess
- AWS CloudFormationFullAccess
- ICH BIN FullAccess
- AWS Lambda\_ FullAccess
- Amazon-API GatewayAdministrator
- Amazon S3 FullAccess
- Amazon EC2 ContainerRegistryFullAccess

Die erste Richtlinie ermöglicht den Zugriff auf Stack-Rollbacks, CloudWatch um Stack-Rollbacks zu aktivieren, wenn ein Alarm auftritt.

Die verbleibenden Richtlinien ermöglichen den AWS SAM Zugriff auf die Dienste und Ressourcen im Stack, die in diesem Tutorial bereitgestellt werden. Weitere Informationen finden Sie unter [Berechtigungen](#) im AWS Serverless Application Model Entwicklerhandbuch.

- h. Wählen Sie Weiter aus.
- i. Geben Sie als Rollenname Folgendes ein:

**codecatalyst-stack-role**

- j. Wählen Sie Rolle erstellen aus.
4. Rufen Sie den ARN der Stack-Rolle wie folgt ab:
    - a. Wählen Sie im Navigationsbereich Rollen aus.
    - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (codecatalyst-stack-role).
    - c. Wählen Sie die Rolle aus der Liste aus.
    - d. Kopieren Sie im Abschnitt Zusammenfassung den ARN-Wert. Sie benötigen sie später.

Sie haben jetzt die Stack-Rolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

### Schritt 3: AWS Rollen hinzufügen zu CodeCatalyst

In diesem Schritt fügen Sie der CodeCatalyst Kontoverbindung in Ihrem Bereich die Build-Rolle (`codecatalyst-build-role`) und die Bereitstellungsrolle (`codecatalyst-deploy-role`) hinzu.

#### Note

Sie müssen die Stack-Rolle (`codecatalyst-stack-role`) nicht zur Verbindung hinzufügen. Das liegt daran, dass die Stack-Rolle von CloudFormation (nicht CodeCatalyst) verwendet wird, nachdem bereits eine Verbindung zwischen CodeCatalyst und AWS mithilfe der Deploy-Rolle hergestellt wurde. Da die Stack-Rolle nicht für den CodeCatalyst Zugriff verwendet wird AWS, muss sie keiner Kontoverbindung zugeordnet werden.

Um Ihrer Kontoverbindung Rollen hinzuzufügen, zu erstellen und bereitzustellen

1. Navigieren CodeCatalyst Sie darin zu Ihrem Bereich.
2. Wählen Sie AWS accounts (-Konten). Eine Liste der Kontoverbindungen wird angezeigt.
3. Wählen Sie die Kontoverbindung aus, die dem AWS Konto entspricht, in dem Sie Ihre Build- und Deploy-Rollen erstellt haben.
4. Wählen Sie in der AWS Managementkonsole die Option Rollen verwalten aus.

Die Seite „IAM-Rolle zu Amazon CodeCatalyst Space hinzufügen“ wird angezeigt. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

5. Wählen Sie Eine bestehende Rolle hinzufügen, die Sie in IAM erstellt haben.

Eine Dropdownliste wird angezeigt. In der Liste werden alle IAM-Rollen mit einer Vertrauensrichtlinie angezeigt, die die Dienstprinzipale `codecatalyst-runner.amazonaws.com` und die `codecatalyst.amazonaws.com` Dienstprinzipale umfasst.

6. Wählen Sie `codecatalyst-build-role` in der Dropdownliste die Option Rolle hinzufügen aus.

7. Wählen Sie IAM-Rolle hinzufügen aus, wählen Sie Vorhandene Rolle hinzufügen, die Sie in IAM erstellt haben, und wählen Sie in der Drop-down-Liste die Option aus. `codecatalyst-deploy-role` Wählen Sie Rolle hinzufügen aus.

Sie haben jetzt die Build- und Deploy-Rollen zu Ihrem Bereich hinzugefügt.

8. Kopieren Sie den Wert des CodeCatalyst Amazon-Anzeigenamens. Sie benötigen diesen Wert später, wenn Sie Ihren Workflow erstellen.

## Schritt 4: Erstellen Sie einen Amazon S3 S3-Bucket

In diesem Schritt erstellen Sie einen Amazon S3 S3-Bucket, in dem Sie die ZIP-Datei des Bereitstellungspakets Ihrer serverlosen Anwendung speichern.

So erstellen Sie einen Amazon-S3-Bucket

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie im Hauptbereich Bucket erstellen aus.
3. Geben Sie als Bucket-Namen Folgendes ein:

```
codecatalyst-cfn-s3-bucket
```

4. Wählen Sie unter AWS -Region eine Region aus. In diesem Tutorial wird davon ausgegangen, dass Sie US West (Oregon) `us-west-2` ausgewählt haben. Informationen zu den von Amazon S3 unterstützten Regionen finden Sie unter [Amazon Simple Storage Service-Endpunkte und Kontingente](#) in der Allgemeine AWS-Referenz.
5. Wählen Sie unten auf der Seite Bucket erstellen aus.

Sie haben jetzt einen Bucket mit dem Namen **codecatalyst-cfn-s3-bucket** in der Region USA West (Oregon) `us-west-2` erstellt.

## Schritt 5: Quelldateien hinzufügen

In diesem Schritt fügen Sie Ihrem CodeCatalyst Quell-Repository mehrere Anwendungsquelldateien hinzu. Der `hello-world` Ordner enthält die Anwendungsdateien, die Sie bereitstellen werden. Der `tests` Ordner enthält Komponententests. Die Ordnerstruktur ist wie folgt:

```
.
```

```
|– hello-world
|  |– tests
|    |– unit
|      |– test-handler.js
|  |– app.js
|– .npmignore
|– package.json
|– sam-template.yml
|– setup-sam.sh
```

## .npmignore-Datei

Die `.npmignore` Datei gibt an, welche Dateien und Ordner npm aus dem Anwendungspaket ausschließen soll. In diesem Tutorial schließt npm den `tests` Ordner aus, da er nicht Teil der Anwendung ist.

Um die Datei `.npmignore` hinzuzufügen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wähle dein Projekt, `codecatalyst-cfn-project`
3. Wählen Sie im Navigationsbereich Code und dann Source Repositories aus.
4. Wählen Sie aus der Liste der Quell-Repositorys Ihr Repository aus. `codecatalyst-cfn-source-repository`
5. Wählen Sie unter Dateien die Option Datei erstellen aus.
6. Geben Sie als Dateiname Folgendes ein:

```
.npmignore
```

7. Geben Sie in das Textfeld den folgenden Code ein:

```
tests/*
```

8. Wählen Sie Commit und anschließend erneut Commit aus.

Sie haben jetzt eine Datei mit dem Namen `.npmignore` im Stammverzeichnis Ihres Repositorys erstellt.



## package.json-Datei

Die `package.json` Datei enthält wichtige Metadaten zu Ihrem Node-Projekt wie Projektname, Versionsnummer, Beschreibung, Abhängigkeiten und andere Details, die beschreiben, wie Sie mit Ihrer Anwendung interagieren und sie ausführen.

Die `package.json` in diesem Tutorial enthaltene Liste enthält eine Liste von Abhängigkeiten und ein `test` Skript. Das Testskript macht Folgendes:

- Unter Verwendung von [Mocha](#) führt das Testskript die in angegebenen Komponententests aus `hello-world/tests/unit/` und schreibt die Ergebnisse mithilfe des [Xunit-Reporters](#) in eine `junit.xml` Datei.
- [Unter Verwendung von Istanbul \(NYC\) generiert das Testskript mithilfe des Clover Reporters einen Bericht über die Codeabdeckung \(`clover.xml`\)](#). Weitere Informationen finden Sie in der [Istanbul-Dokumentation unter Verwenden alternativer Reporter](#).

Um die Datei `package.json` hinzuzufügen

1. Wählen Sie in Ihrem Repository unter Dateien die Option Datei erstellen aus.
2. Geben Sie als Dateiname Folgendes ein:

```
package.json
```

3. Geben Sie in das Textfeld den folgenden Code ein:

```
{
  "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. Wählen Sie Commit und anschließend erneut Commit aus.

Sie haben jetzt eine Datei mit dem Namen `package.json` zum Stammverzeichnis des Repositorys hinzugefügt.

#### Datei `sam-template.yml`

Die `sam-template.yml` Datei enthält die Anweisungen für die Bereitstellung der Lambda-Funktion und des API Gateway und deren gemeinsame Konfiguration. Sie folgt der [AWS Serverless Application Model Vorlagenspezifikation](#), die die AWS CloudFormation Vorlagenspezifikation erweitert.

In diesem Tutorial verwenden Sie eine AWS SAM Vorlage anstelle einer regulären AWS CloudFormation Vorlage, da sie einen hilfreichen [AWS::Serverless::Function](#) Ressourcentyp AWS SAM bietet. Dieser Typ führt viele behind-the-scenes Konfigurationen durch, die Sie normalerweise ausschreiben müssen, um die grundlegende Syntax zu verwenden. CloudFormation Beispielsweise `AWS::Serverless::Function` erstellt der eine Lambda-Funktion, eine Lambda-Ausführungsrolle und Ereignisquellenzuordnungen, mit denen die Funktion gestartet wird. Sie müssen all das programmieren, wenn Sie es mit Basic schreiben möchten. CloudFormation

In diesem Tutorial wird zwar eine vorgefertigte Vorlage verwendet, Sie können jedoch eine Vorlage als Teil Ihres Workflows mithilfe einer Build-Aktion generieren. Weitere Informationen finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#).

Um die Datei `sam-template.yml` hinzuzufügen

1. Wählen Sie in Ihrem Repository unter Dateien die Option Datei erstellen aus.
2. Geben Sie als Dateiname Folgendes ein:

```
sam-template.yml
```

### 3. Geben Sie in das Textfeld den folgenden Code ein:

```
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/awslabs/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/awslabs/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. Wählen Sie Commit und anschließend erneut Commit aus.

Sie haben jetzt eine Datei mit `sam-template.yml` dem Namen im Stammordner Ihres Repositories hinzugefügt.

Datei `setup-sam.sh`

Die `setup-sam.sh` Datei enthält Anweisungen zum Herunterladen und Installieren des AWS SAM CLI-Dienstprogramms. Der Workflow verwendet dieses Hilfsprogramm, um die `hello-world` Quelle zu verpacken.

Um die Datei `setup-sam.sh` hinzuzufügen

1. Wählen Sie in Ihrem Repository unter Dateien die Option Datei erstellen aus.
2. Geben Sie als Dateiname Folgendes ein:

```
setup-sam.sh
```

3. Geben Sie in das Textfeld den folgenden Code ein:

```
#!/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
```

Ersetzen Sie im vorherigen Code `us-west-2` durch Ihre AWS Region.

4. Wählen Sie Commit und anschließend erneut Commit aus.

Sie haben jetzt eine Datei mit dem Namen `setup-sam.sh` zum Stammverzeichnis des Repositoriums hinzugefügt.

## Datei `app.js`

Das `app.js` enthält den Lambda-Funktionscode. In diesem Tutorial gibt der Code den Text `hello world` zurück.

Um die Datei `app.js` hinzuzufügen

1. Wählen Sie in Ihrem Repository unter Dateien die Option Datei erstellen aus.
2. Geben Sie als Dateiname Folgendes ein:

```
hello-world/app.js
```

3. Geben Sie in das Textfeld den folgenden Code ein:

```
// 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. Wählen Sie Commit und anschließend erneut Commit aus.

Sie haben jetzt einen Ordner mit dem Namen `hello-world` und eine Datei mit dem Namen `erstelltapp.js`.

Datei `test-handler.js`

Die `test-handler.js` Datei enthält Komponententests für die Lambda-Funktion.

Um die Datei `test-handler.js` hinzuzufügen

1. Wählen Sie in Ihrem Repository unter Dateien die Option Datei erstellen aus.
2. Geben Sie als Dateiname Folgendes ein:

```
hello-world/tests/unit/test-handler.js
```

3. Geben Sie in das Textfeld den folgenden Code ein:

```
'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).toBe.an('object');
    expect(result.statusCode).toEqual(200);
    expect(result.body).toBe.an('string');

    let response = JSON.parse(result.body);

    expect(response).toBe.an('object');
    expect(response.message).toEqual("hello world");
    // expect(response.location).toBe.an("string");
  });
});
```

#### 4. Wählen Sie Commit und anschließend erneut Commit aus.

Sie haben jetzt eine Datei mit `test-handler.js` dem Namen unter dem `hello-world/tests/unit` Ordner hinzugefügt.

Sie haben jetzt alle Ihre Quelldateien hinzugefügt.

Nehmen Sie sich einen Moment Zeit, um Ihre Arbeit zu überprüfen und sicherzustellen, dass Sie alle Dateien in den richtigen Ordnern abgelegt haben. Die Ordnerstruktur ist wie folgt:

```
.
|- hello-world
|  |- tests
|    |- unit
|      |- test-handler.js
|  |- app.js
|- .npmignore
|- README.md
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

## Schritt 6: Erstellen Sie einen Workflow und führen Sie ihn aus

In diesem Schritt erstellen Sie einen Workflow, der Ihren Lambda-Quellcode verpackt und bereitstellt. Der Workflow besteht aus den folgenden Bausteinen, die sequentiell ausgeführt werden:

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).

- Eine Testaktion (Test) — Beim Trigger installiert diese Aktion den [Node-Paketmanager \(npm\)](#) und führt dann den `npm run test` Befehl aus. Dieser Befehl weist npm an, das in der `test` Datei definierte Skript auszuführen. `package.json` Das `test` Skript wiederum führt die Komponententests aus und generiert zwei Berichte: einen Testbericht (`junit.xml`) und einen Bericht über die Codeabdeckung (`clover.xml`). Weitere Informationen finden Sie unter [package.json-Datei](#).

Als Nächstes wandelt die Testaktion die XML-Berichte in CodeCatalyst Berichte um und zeigt sie in der CodeCatalyst Konsole auf der Registerkarte Berichte der Testaktion an.

Weitere Informationen zur Testaktion finden Sie unter [Testen mit Workflows in CodeCatalyst](#).

- Eine Build-Aktion (BuildBackend) — Nach Abschluss der Testaktion lädt die Build-Aktion die AWS SAM CLI herunter und installiert sie, packt die `hello-world` Quelle und kopiert das Paket in Ihren Amazon S3 S3-Bucket, wo es vom Lambda-Service erwartet wird. Die Aktion gibt auch eine neue AWS SAM Vorlagendatei namens `aws-sam-template-packaged.yml` und platziert sie in einem Ausgabeartefakt namens `buildArtifact`

Weitere Informationen zur Build-Aktion finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#).

- Eine Bereitstellungsaktion (DeployCloudFormationStack) — Nach Abschluss der Build-Aktion sucht die Bereitstellungsaktion nach dem von der Build-Aktion (`buildArtifact`) generierten Ausgabeartefakt, findet die darin enthaltene AWS SAM Vorlage und führt dann die Vorlage aus. Die AWS SAM Vorlage erstellt einen Stack, der die serverlose Anwendung bereitstellt.

So erstellen Sie ein Workflow

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie Workflow erstellen aus.
3. Wählen Sie für Quell-Repository die Option `codecatalyst-cfn-source-repository`.
4. Wählen Sie für Branch die Option `main`.
5. Wählen Sie Erstellen.
6. Löschen Sie den YAML-Beispielcode.
7. Fügen Sie den folgenden YAML-Code hinzu:

```
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
```

Ersetzen Sie im vorherigen Code:

- Beide Instanzen von *codecatalyst-cfn-environment* mit dem Namen Ihrer Umgebung.
- Beide Instanzen von *codecatalyst-account-connection* mit dem Anzeigenamen Ihrer Kontoverbindung. Der Anzeigename kann eine Zahl sein. Weitere Informationen finden Sie unter [Schritt 3: AWS Rollen hinzufügen zu CodeCatalyst](#).
- *codecatalyst-build-role* mit dem Namen der Build-Rolle, in der Sie erstellt haben [Schritt 2: AWS Rollen erstellen](#).
- *codecatalyst-cfn-s3-Bucket* mit dem Namen des Amazon S3 S3-Buckets, in [Schritt 4: Erstellen Sie einen Amazon S3 S3-Bucket](#) dem Sie ihn erstellt haben.
- Beide Instances von *us-west-2* mit der Region, in der sich Ihr Amazon S3 S3-Bucket befindet (erste Instance) und in der Ihr Stack bereitgestellt wird (zweite Instance). Diese Regionen können unterschiedlich sein. In diesem Tutorial wird davon ausgegangen, dass beide Regionen auf eingestellt sind *us-west-2*. Einzelheiten zu den von Amazon S3 und AWS CloudFormation unterstützten Regionen finden Sie unter [Service-Endpunkte und Kontingente](#) in der Allgemeine AWS-Referenz.

- `codecatalyst-deploy-role` mit dem Namen der Bereitstellungsrolle, in [Schritt 2: AWS Rollen erstellen](#) der Sie erstellt haben.
- `codecatalyst-cfn-environment` mit dem Namen der Umgebung, in der Sie erstellt haben [Voraussetzungen](#).
- `arn:aws:iam: :111122223333:role/ StackRole` mit dem Amazon-Ressourcennamen (ARN) der Stack-Rolle, in der Sie erstellt haben. [Schritt 2: AWS Rollen erstellen](#)

 Note

Wenn Sie sich entschieden haben, keine Build-, Deploy- und Stack-Rollen zu erstellen, ersetzen Sie `codecatalyst-build-role` `codecatalyst-deploy-role`, und `arn:aws:iam: StackRole :111122223333:role/` durch den Namen oder ARN der Rolle. `CodeCatalystWorkflowDevelopmentRole-spaceName` Weitere Informationen über diese Rolle finden Sie unter [Schritt 2: AWS Rollen erstellen](#).

Informationen zu den Eigenschaften des zuvor gezeigten Codes finden Sie unter [Referenz zur Aktion „Stack bereitstellen AWS CloudFormation“](#)

8. (Optional) Wählen Sie „Validieren“, um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie ihn festschreiben.
9. Wählen Sie Commit (Übergeben).
10. Geben Sie im Dialogfeld „Workflow bestätigen“ Folgendes ein:
  - a. Behalten Sie als Workflow-Dateiname die Standardeinstellung `codecatalyst-cfn-workflow`.
  - b. Geben Sie für Commit-Nachricht Folgendes ein:

```
add initial workflow file
```

- c. Wählen Sie für Repository `codecatalyst-cfn-source-repository`.
- d. Wählen Sie als Branch-Name `main` aus.
- e. Wählen Sie Commit (Übergeben).

Sie haben jetzt einen Workflow erstellt. Eine Workflow-Ausführung wird aufgrund des oben im Workflow definierten Triggers automatisch gestartet. Insbesondere, als Sie die `codecatalyst-`

`cfn-workflow.yaml` Datei in Ihr Quell-Repository übernommen (und per Push übertragen) haben, hat der Trigger die Workflow-Ausführung gestartet.

Um den laufenden Workflow-Lauf zu sehen

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie den Workflow aus, den Sie gerade erstellt haben: `codecatalyst-cfn-workflow`
3. Wählen Sie die Registerkarte „Läufe“.
4. Wählen Sie in der Spalte Run-ID die Run-ID aus.
5. Wählen Sie Test, um den Fortschritt der Tests zu sehen.
6. Wählen Sie BuildBackend, um den Baufortschritt zu sehen.
7. Wählen Sie DeployCloudFormationStack diese Option, um den Fortschritt der Bereitstellung zu sehen.

Weitere Informationen zum Anzeigen von Ausführungsdetails finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).

8. Wenn die DeployCloudFormationStackAktion abgeschlossen ist, gehen Sie wie folgt vor:
  - Wenn die Workflow-Ausführung erfolgreich war, fahren Sie mit dem nächsten Verfahren fort.
  - Wenn die Workflow-Ausführung beim Test oder bei der BuildBackendAktion fehlgeschlagen ist, wählen Sie Protokolle aus, um das Problem zu beheben.
  - Wenn die Workflow-Ausführung bei der DeployCloudFormationStackAktion fehlgeschlagen ist, wählen Sie die Aktion Bereitstellen und dann die Registerkarte Zusammenfassung aus. Scrollen Sie zum Abschnitt CloudFormation Ereignisse, um die detaillierte Fehlermeldung anzuzeigen. Wenn ein Rollback aufgetreten ist, löschen Sie den `codecatalyst-cfn-stack` Stack über die AWS CloudFormation Konsole, AWS bevor Sie den Workflow erneut ausführen.

Um die Bereitstellung zu überprüfen

1. Wählen Sie nach einer erfolgreichen Bereitstellung in der horizontalen Menüleiste oben die Option Variablen (7) aus. (Wählen Sie im Bereich auf der rechten Seite nicht Variablen aus.)
2. Fügen Sie HelloWorldApials Nächstes die `https://` URL in einen Browser ein.

Eine Hello-World-JSON-Meldung von der Lambda-Funktion wird angezeigt, die darauf hinweist, dass der Workflow die Lambda-Funktion und das API Gateway erfolgreich bereitgestellt und konfiguriert hat.

 Tip

Sie können diese URL mit einigen kleinen Konfigurationen im Workflow-Diagramm CodeCatalyst anzeigen lassen. Weitere Informationen finden Sie unter [Anzeige der URL der bereitgestellten Anwendung](#).

Um die Ergebnisse der Komponententests und die Codeabdeckung zu überprüfen

1. Wählen Sie im Workflow-Diagramm Test und dann Berichte aus.
2. Wählen TestReportSie, ob Sie sich die Ergebnisse der Komponententests oder die Codeabdeckung der getesteten Dateien anzeigen lassen möchten CoverageReport, in diesem Fall `app.js` und `test-handler.js`.

Um die bereitgestellten Ressourcen zu überprüfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die API Gateway Gateway-Konsole unter <https://console.aws.amazon.com/apigateway/>.
2. Beachten Sie die `codecatalyst-cfn-stackAPI`, die die AWS SAM Vorlage erstellt hat. Der API-Name stammt aus dem `Configuration/name` Wert in der Workflow-Definitionsdatei (`codecatalyst-cfn-workflow.yaml`).
3. Öffnen Sie die AWS Lambda Konsole unter <https://console.aws.amazon.com/lambda/>.
4. Wählen Sie im Navigationsbereich Funktionen aus.
5. Wählen Sie Ihre Lambda-Funktion, `codecatalyst-cfn-stack-HelloWorldFunction-string`.
6. Sie können sehen, wie das API Gateway ein Auslöser für die Funktion ist. Diese Integration wurde automatisch nach dem AWS SAM `AWS::Serverless::Function` Ressourcentyp konfiguriert.

## Schritt 7: Nehmen Sie eine Änderung vor

In diesem Schritt nehmen Sie eine Änderung an Ihrem Lambda-Quellcode vor und übernehmen diese. Mit diesem Commit wird ein neuer Workflow-Lauf gestartet. Bei diesem Lauf wird die neue Lambda-Funktion in einem blaugrünen Schema bereitgestellt, das die in der Lambda-Konsole angegebene Standardkonfiguration zur Verkehrsverlagerung verwendet.

Um eine Änderung an Ihrer Lambda-Quelle vorzunehmen

1. Navigieren Sie in CodeCatalyst zu Ihrem Projekt.
2. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.
3. Wählen Sie Ihr Quell-Repository `codecatalyst-cfn-source-repository` aus.
4. Ändern Sie die Anwendungsdatei:
  - a. Wählen Sie den Ordner `hello-world` aus.
  - b. Wählen Sie die `app.js` Datei aus.
  - c. Wählen Sie Bearbeiten aus.
  - d. Wechseln Sie in Zeile 23 `hello world` zu **Tutorial complete!**.
  - e. Wählen Sie Commit und anschließend erneut Commit aus.

Durch den Commit wird ein Workflow-Lauf gestartet. Dieser Lauf schlägt fehl, weil Sie die Komponententests nicht aktualisiert haben, um die Namensänderung widerzuspiegeln.

5. Aktualisieren Sie die Komponententests:
  - a. Wählen Sie `hello-world\tests\unit\test-handler.js`.
  - b. Wählen Sie Bearbeiten aus.
  - c. Wechseln Sie in Zeile 19 `hello world` zu **Tutorial complete!**.
  - d. Wählen Sie Commit und anschließend erneut Commit aus.

Durch den Commit wird ein weiterer Workflow-Lauf gestartet. Dieser Lauf wird erfolgreich sein.

6. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
7. Wählen Sie **codecatalyst-cfn-workflow** und wählen Sie dann Runs aus.
8. Wählen Sie die Lauf-ID des letzten Laufs aus. Es sollte noch in Bearbeitung sein.
9. Wählen Sie Test und BuildBackend, DeployCloudFormationStackum den Fortschritt der Workflow-Ausführung zu sehen.

10. Wenn der Workflow abgeschlossen ist, wählen Sie oben Variablen (7) aus.
11. Fügen Sie HelloWorldApi als Nächstes die `https://` URL in einen Browser ein.

Im Browser wird eine `Tutorial complete!` Meldung angezeigt, die darauf hinweist, dass Ihre neue Anwendung erfolgreich bereitgestellt wurde.

## Bereinigen

Bereinigen Sie die in diesem Tutorial verwendeten Dateien und Dienste, um zu vermeiden, dass dafür Gebühren anfallen.

Um in der CodeCatalyst Konsole aufzuräumen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Löschen Sie `codecatalyst-cfn-workflow`.
3. Löschen Sie `codecatalyst-cfn-environment`.
4. Löschen Sie `codecatalyst-cfn-source-repository`.
5. Löschen Sie `codecatalyst-cfn-project`.

Zum Aufräumen in der AWS Management Console

1. Aufräumen CloudFormation, wie folgt:
  - a. Öffnen Sie die AWS CloudFormation Konsole unter <https://console.aws.amazon.com/cloudformation>.
  - b. Löschen Sie das `codecatalyst-cfn-stack`.

Durch das Löschen des Stacks werden alle Tutorial-Ressourcen aus den API Gateway- und Lambda-Diensten entfernt.

2. Bereinigen Sie in Amazon S3 wie folgt:
  - a. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
  - b. Wählen Sie das Symbol `codecatalyst-cfn-s3-bucket`.
  - c. Löschen Sie den Inhalt des Buckets.
  - d. Löschen Sie den Bucket.
3. Bereinigen Sie in IAM wie folgt:

- a. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- b. Löschen Sie das `codecatalyst-deploy-policy`.
- c. Löschen Sie das `codecatalyst-build-policy`.
- d. Löschen Sie das `codecatalyst-stack-policy`.
- e. Löschen Sie das `codecatalyst-deploy-role`.
- f. Löschen Sie das `codecatalyst-build-role`.
- g. Löschen Sie das `codecatalyst-stack-role`.

In diesem Tutorial haben Sie gelernt, wie Sie eine serverlose Anwendung mithilfe eines CodeCatalyst Workflows und einer CloudFormation Stack-Aktion bereitstellen als AWS CloudFormation Stack bereitstellen.

## Tutorial: Bereitstellen einer Anwendung in Amazon ECS

In diesem Tutorial erfahren Sie, wie Sie mithilfe eines Workflows, Amazon ECS und einiger anderer AWS Services eine serverlose Anwendung in Amazon Elastic Container Service (Amazon ECS) bereitstellen. Bei der bereitgestellten Anwendung handelt es sich um eine einfache Hello World-Website, die auf einem Docker-Image des Apache-Webserver basiert. Das Tutorial führt Sie durch die erforderlichen Vorbereitungsarbeiten, z. B. die Einrichtung eines Clusters, und beschreibt anschließend, wie Sie einen Workflow für die Erstellung und Bereitstellung der Anwendung erstellen.

### Tip

Anstatt sich durch dieses Tutorial zu arbeiten, können Sie einen Blueprint verwenden, der ein vollständiges Amazon ECS-Setup für Sie durchführt. Sie müssen entweder die Node.js API mit AWS Fargate oder die Java API mit AWS Fargate Blueprint verwenden. Weitere Informationen finden Sie unter [Ein Projekt mit einem Blueprint erstellen](#).

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Richten Sie einen AWS Benutzer ein und AWS CloudShell](#)
- [Schritt 2: Stellen Sie eine Platzhalteranwendung in Amazon ECS bereit](#)
- [Schritt 3: Erstellen Sie ein Amazon ECR-Image-Repository](#)



- [Schritt 4: AWS Rollen erstellen](#)
- [Schritt 5: AWS Rollen hinzufügen CodeCatalyst](#)
- [Schritt 6: Erstellen Sie ein Quell-Repository](#)
- [Schritt 7: Quelldateien hinzufügen](#)
- [Schritt 8: Einen Workflow erstellen und ausführen](#)
- [Schritt 9: Nehmen Sie eine Änderung an Ihren Quelldateien vor](#)
- [Bereinigen](#)

## Voraussetzungen

Bevor Sie beginnen:

- Sie benötigen einen CodeCatalyst Bereich mit einem verbundenen AWS Konto. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).
- In deinem Space benötigst du ein leeres CodeCatalyst Projekt mit dem Namen:

```
codecatalyst-ecs-project
```

Weitere Informationen finden Sie unter [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#).

- In Ihrem Projekt benötigen Sie eine CodeCatalyst Umgebung namens:

```
codecatalyst-ecs-environment
```

Konfigurieren Sie diese Umgebung wie folgt:

- Wählen Sie einen beliebigen Typ aus, z. B. „Keine Produktion“.
- Connect dein AWS Konto damit.

Weitere Informationen finden Sie unter [Arbeiten mit Umgebungen](#).

## Schritt 1: Richten Sie einen AWS Benutzer ein und AWS CloudShell

Der erste Schritt in diesem Tutorial besteht darin AWS IAM Identity Center, einen Benutzer zu erstellen und eine AWS CloudShell Instanz als dieser Benutzer zu starten. Für die Dauer dieses Tutorials CloudShell ist dies Ihr Entwicklungscomputer, auf dem Sie AWS Ressourcen und Dienste konfigurieren. Löschen Sie diesen Benutzer, nachdem Sie das Tutorial abgeschlossen haben.

**Note**

Verwenden Sie für dieses Tutorial nicht Ihren Root-Benutzer. Sie müssen einen separaten Benutzer erstellen, da sonst später Probleme bei der Ausführung von Aktionen in der AWS Command Line Interface (CLI) auftreten können.

Weitere Informationen zu IAM Identity Center-Benutzern finden Sie im AWS IAM Identity Center Benutzerhandbuch und im AWS CloudShell Benutzerhandbuch. CloudShell

So erstellen Sie einen IAM Identity Center-Benutzer

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS IAM Identity Center Konsole unter <https://console.aws.amazon.com/singlesignon/>.

**Note**

Stellen Sie sicher, dass Sie sich mit dem anmelden AWS-Konto , der mit Ihrem CodeCatalyst Bereich verbunden ist. Sie können überprüfen, welches Konto verbunden ist, indem Sie zu Ihrem Bereich navigieren und den Tab AWS-Konten auswählen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).

2. Wählen Sie im Navigationsbereich Users (Benutzer) und dann Add User (Benutzer hinzufügen) aus.
3. Geben Sie im Feld Nutzernamen Folgendes ein:

**CodeCatalystECSUser**

4. Wählen Sie unter Passwort die Option Einmalpasswort generieren aus, das Sie mit diesem Benutzer teilen können.
5. Geben Sie in den Feldern E-Mail-Adresse und E-Mail-Adresse bestätigen eine E-Mail-Adresse ein, die noch nicht in IAM Identity Center existiert.
6. Geben Sie in die Felder Vorname und Nachname Folgendes ein:

**CodeCatalystECSUser**

7. Behalten Sie im Feld Anzeigenamen den automatisch generierten Namen bei:

**CodeCatalystECSUser CodeCatalystECSUser**

8. Wählen Sie Weiter aus.
9. Wählen Sie auf der Seite Benutzer zu Gruppen hinzufügen die Option Weiter aus.
10. Überprüfen Sie auf der Seite Benutzer überprüfen und hinzufügen die Informationen und wählen Sie Benutzer hinzufügen aus.

Ein Dialogfeld mit einem Einmalkennwort wird angezeigt.

11. Wählen Sie Kopieren und fügen Sie dann die Anmeldeinformationen ein, einschließlich der URL des AWS Zugriffsportals und des Einmalkennworts.
12. Klicken Sie auf Schließen.

So erstellen Sie einen Berechtigungssatz

Sie weisen diesen Berechtigungssatz einem CodeCatalystECSUser späteren Zeitpunkt zu.

1. Wählen Sie im Navigationsbereich die Option Berechtigungssätze und dann Berechtigungssatz erstellen aus.
2. Wählen Sie Vordefinierter Berechtigungssatz und dann aus AdministratorAccess. Diese Richtlinie gewährt allen volle Berechtigungen AWS-Services.
3. Wählen Sie Weiter aus.
4. Geben Sie im Feld Name des Berechtigungssatzes Folgendes ein:

**CodeCatalystECSPermissionSet**

5. Wählen Sie Weiter aus.
6. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen und wählen Sie Erstellen aus.

Um den Berechtigungssatz CodeCatalyst ecsUser zuzuweisen


1. Wählen Sie im Navigationsbereich das Kontrollkästchen neben dem aus AWS-Konten, bei dem Sie gerade angemeldet sind AWS-Konto , und aktivieren Sie es anschließend.
2. Wählen Sie Benutzer oder Gruppen zuweisen aus.
3. Wählen Sie die Registerkarte Users.

4. Aktivieren Sie das Kontrollkästchen neben `CodeCatalystECSUser`.
5. Wählen Sie Weiter aus.
6. Aktivieren Sie das Kontrollkästchen neben `CodeCatalystECSPermissionSet`.
7. Wählen Sie Weiter aus.
8. Überprüfen Sie die Informationen und wählen Sie Senden aus.

Sie haben sie nun `CodeCatalystECSPermissionSet` zugewiesen `CodeCatalystECSUser` und an Sie AWS-Konto gebunden.

Um sich ab- und wieder als CodeCatalyst ecsUser anzumelden

1. Bevor Sie sich abmelden, stellen Sie sicher, dass Sie die URL des AWS Zugriffsportals sowie den Benutzernamen und das Einmalpasswort für haben. `CodeCatalystECSUser` Sie hätten diese Informationen früher in einen Texteditor kopieren sollen.

 Note

Wenn Ihnen diese Informationen nicht vorliegen, rufen Sie die `CodeCatalystECSUser` Detailseite im IAM Identity Center auf und wählen Sie Passwort zurücksetzen, Einmalpasswort generieren [...], und klicken Sie erneut auf Passwort zurücksetzen, um die Informationen auf dem Bildschirm anzuzeigen.

2. Melden Sie sich ab AWS.
3. Fügen Sie die URL des AWS Zugangsportals in die Adressleiste Ihres Browsers ein.
4. Melden Sie sich mit dem Benutzernamen und dem Einmalpasswort für `anCodeCatalystECSUser`.
5. Geben Sie unter Neues Passwort ein Passwort ein und wählen Sie Neues Passwort festlegen aus.

Auf dem Bildschirm erscheint ein AWS-KontoFeld.

6. Wählen Sie AWS-Kontound wählen Sie dann den Namen des Benutzers und des AWS-Konto Berechtigungssatzes aus, dem Sie den `CodeCatalystECSUser` Benutzer zugewiesen haben.
7. Wählen Sie neben dem `CodeCatalystECSPermissionSet` die Option Verwaltungskonsole aus.

Das AWS Management Console erscheint. Sie sind jetzt CodeCatalystECSUser mit den entsprechenden Berechtigungen angemeldet.

Um eine AWS CloudShell Instance zu starten

1. Wählen Sie wie CodeCatalystECSUser in der oberen Navigationsleiste das AWS Symbol




Die Hauptseite von wird AWS Management Console angezeigt.

2. Wählen Sie in der oberen Navigationsleiste das AWS CloudShell Symbol



CloudShell öffnet. Warten Sie, bis die CloudShell Umgebung erstellt ist.

 Note

Wenn Sie das CloudShell Symbol nicht sehen, vergewissern Sie sich, dass Sie sich in einer [Region befinden, die von unterstützt wird CloudShell](#). In diesem Tutorial wird davon ausgegangen, dass Sie sich in der Region USA West (Oregon) befinden.

Um zu überprüfen, ob der installiert AWS CLI ist

1. Geben Sie im CloudShell Terminal Folgendes ein:

```
aws --version
```

2. Vergewissern Sie sich, dass eine Version angezeigt wird.

Der AWS CLI ist bereits für den aktuellen Benutzer konfiguriertCodeCatalystECSUser, sodass keine AWS CLI Schlüssel und Anmeldeinformationen konfiguriert werden müssen, wie dies normalerweise der Fall ist.

## Schritt 2: Stellen Sie eine Platzhalteranwendung in Amazon ECS bereit

In diesem Abschnitt stellen Sie manuell eine Platzhalteranwendung in Amazon ECS bereit. Diese Platzhalteranwendung wird durch die Hello World-Anwendung ersetzt, die in Ihrem Workflow bereitgestellt wird. Die Platzhalteranwendung ist Apache Web Server.

Weitere Informationen zu Amazon ECS finden Sie im Amazon Elastic Container Service Developer Guide.

Führen Sie die folgenden Schritte durch, um die Platzhalteranwendung bereitzustellen.

Um die Rolle zur Aufgabenausführung zu erstellen

Diese Rolle gewährt Amazon ECS die AWS Fargate (Fargate) Erlaubnis, API-Aufrufe in Ihrem Namen durchzuführen.

1. Erstellen Sie eine Vertrauensrichtlinie:
  - a. Geben Sie in AWS CloudShell den folgenden Befehl ein:

```
cat > codecatalyst-ecs-trust-policy.json
```

Im CloudShell Terminal erscheint eine blinkende Aufforderung.

- b. Geben Sie an der Eingabeaufforderung den folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- c. Platzieren Sie den Cursor hinter der letzten geschweiften Klammer (}).
    - d. Drücken Sie dann **Enter** und **Ctrl+d**, um die Datei zu speichern und cat zu beenden.

## 2. Erstellen Sie eine Rolle zur Aufgabenausführung:

```
aws iam create-role \  
  --role-name codecatalyst-ecs-task-execution-role \  
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

## 3. Hängen Sie die AWS verwaltete AmazonECSTaskExecutionRolePolicy Richtlinie an die Rolle an:

```
aws iam attach-role-policy \  
  --role-name codecatalyst-ecs-task-execution-role \  
  --policy-arn arn:aws:iam::aws:policy/service-role/  
AmazonECSTaskExecutionRolePolicy
```

## 4. Die Details der Rolle anzeigen:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-task-execution-role
```

## 5. Notieren Sie sich den "Arn": Wert der Rolle, zum Beispiel `arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role`. Sie benötigen diesen Amazon-Ressourcennamen (ARN) später.

## Erstellen eines Amazon ECS-Clusters

Dieser Cluster wird die Apache-Platzhalteranwendung und später die Hello World-Anwendung enthalten.

### 1. Wie in CodeCatalystECSUser AWS CloudShell, erstellen Sie einen leeren Cluster:

```
aws ecs create-cluster --cluster-name codecatalyst-ecs-cluster
```

### 2. (Optional) Stellen Sie sicher, dass der Cluster erfolgreich erstellt wurde:

```
aws ecs list-clusters
```

Der ARN des `codecatalyst-ecs-cluster` Clusters sollte in der Liste erscheinen, was auf eine erfolgreiche Erstellung hinweist.

## Um eine Aufgabendefinitionsdatei zu erstellen

Die Aufgabendefinitionsdatei gibt an, dass das Docker-Image (httpd:2.4) des [Apache 2.4-Webservers](#) ausgeführt werden soll, von DockerHub dem abgerufen wird.

1. Wie in CodeCatalystECSUser AWS CloudShell, erstellen Sie eine Aufgabendefinitionsdatei:

```
cat > taskdef.json
```

2. Fügen Sie den folgenden Code an der Eingabeaufforderung ein:

```
{
  "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"
}
```

Ersetzen Sie im vorherigen Code `arn:aws:iam::111122223333:role/`-role codecatalyst-ecs-task-execution

mit dem ARN der Aufgabenausführungsrolle, die Sie notiert haben [Um die Rolle zur Aufgabenausführung zu erstellen](#).



3. Platzieren Sie den Cursor hinter der letzten geschweiften Klammer (}).
4. Drücken Sie dann **Enter** und **Ctrl+d**, um die Datei zu speichern und cat zu beenden.

Um die Aufgabendefinitionsdatei bei Amazon ECS zu registrieren

1. Wie in CodeCatalystECSUser AWS CloudShell, registrieren Sie die Aufgabendefinition:

```
aws ecs register-task-definition \  
  --cli-input-json file://taskdef.json
```

2. (Optional) Stellen Sie sicher, dass die Aufgabendefinition registriert wurde:

```
aws ecs list-task-definitions
```

Die `codecatalyst-ecs-task-def` Aufgabendefinition sollte in der Liste erscheinen.

Um den Amazon ECS-Service zu erstellen

Der Amazon ECS-Service führt die Aufgaben (und die zugehörigen Docker-Container) der Apache-Platzhalteranwendung und später der Hello World-Anwendung aus.

1. Wechseln Sie außerdem zur Amazon Elastic Container Service-Konsole, falls Sie dies noch nicht getan haben. CodeCatalystECSUser
2. Wählen Sie den Cluster aus, den Sie zuvor erstellt haben, `codecatalyst-ecs-cluster`.
3. Wählen Sie auf der Registerkarte Dienste die Option Erstellen aus.
4. Gehen Sie auf der Seite Erstellen wie folgt vor:
  - a. Behalten Sie alle Standardeinstellungen mit Ausnahme der im Folgenden aufgeführten bei.
  - b. Wählen Sie unter Launch type (Starttyp) FARGATE aus.
  - c. Wählen Sie unter Aufgabendefinition in der Dropdownliste Familie die folgenden Optionen aus:  
  
`codecatalyst-ecs-task-def`
  - d. Geben Sie als Dienstname Folgendes ein:


```
codecatalyst-ecs-service
```

- e. Geben Sie für Gewünschte Aufgaben Folgendes ein:


3

In diesem Tutorial startet jede Aufgabe einen einzelnen Docker-Container.

- f. Erweitern Sie den Bereich Netzwerk.  
g. Wählen Sie für VPC eine beliebige VPC aus.  
h. Wählen Sie für Subnetze ein beliebiges Subnetz aus.

 Note

Geben Sie nur ein Subnetz an. Das ist alles, was für dieses Tutorial benötigt wird.

 Note

Wenn Sie keine VPC und kein Subnetz haben, erstellen Sie sie. Weitere Informationen finden [Sie unter Erstellen einer VPC](#) und [Erstellen eines Subnetzes in Ihrer VPC](#) im Amazon VPC-Benutzerhandbuch.

- i. Wählen Sie für Sicherheitsgruppe die Option Create a new security group aus und gehen Sie dann wie folgt vor:
- i. Geben Sie als Namen der Sicherheitsgruppe Folgendes ein:

**codecatalyst-ecs-security-group**

- ii. Geben Sie als Beschreibung der Sicherheitsgruppe Folgendes ein:

**CodeCatalyst ECS security group**

- iii. Wählen Sie Regel hinzufügen aus. Wählen Sie für Typ die Option HTTP und für Quelle die Option Anywhere aus.  
j. Wählen Sie unten Create aus.  
k. Warten Sie, bis der Dienst erstellt wurde. Dies kann einige Minuten dauern.
5. Wählen Sie die Registerkarte Aufgaben und dann die Schaltfläche „Aktualisieren“. Vergewissern Sie sich, dass die Spalte Letzter Status für alle drei Aufgaben auf Wird ausgeführt gesetzt ist.

(Optional) Um zu überprüfen, ob Ihre Apache-Platzhalteranwendung ausgeführt wird

1. Wählen Sie auf der Registerkarte Aufgaben eine der drei Aufgaben aus.
2. Wählen Sie im Feld Öffentliche IP die Option Open Address aus.

Eine `It Works!` Seite wird angezeigt. Dies weist darauf hin, dass der Amazon ECS-Service erfolgreich eine Aufgabe gestartet hat, die einen Docker-Container mit dem Apache-Image gestartet hat.

An dieser Stelle des Tutorials haben Sie manuell einen Amazon ECS-Cluster, eine Service- und Aufgabendefinition sowie eine Apache-Platzhalteranwendung bereitgestellt. Nachdem Sie all diese Elemente eingerichtet haben, sind Sie nun bereit, einen Workflow zu erstellen, der die Apache-Platzhalteranwendung durch die Hello World-Anwendung des Tutorials ersetzt.

### Schritt 3: Erstellen Sie ein Amazon ECR-Image-Repository

In diesem Abschnitt erstellen Sie ein privates Image-Repository in Amazon Elastic Container Registry (Amazon ECR). In diesem Repository wird das Docker-Image des Tutorials gespeichert, das das zuvor bereitgestellte Apache-Platzhalter-Image ersetzt.

Weitere Informationen zu Amazon ECR finden Sie im Amazon Elastic Container Registry User Guide.

Um ein Bild-Repository in Amazon ECR zu erstellen

1. Wie in `CodeCatalystECSUser` AWS CloudShell, ein leeres Repository in Amazon ECR erstellen:

```
aws ecr create-repository --repository-name codecatalyst-ecs-image-repo
```

2. Zeigen Sie die Details des Amazon ECR-Repositorys an:

```
aws ecr describe-repositories \  
  --repository-names codecatalyst-ecs-image-repo
```

3. Notieren Sie sich den `repositoryUri`: Wert, zum Beispiel. `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

Sie benötigen ihn später, wenn Sie das Repository zu Ihrem Workflow hinzufügen.

## Schritt 4: AWS Rollen erstellen

In diesem Abschnitt erstellen Sie AWS IAM-Rollen, die Ihr CodeCatalyst Workflow benötigt, um zu funktionieren. Diese Rollen sind:

- Build-Rolle — Erteilt der CodeCatalyst Build-Aktion (im Workflow) die Berechtigung, auf Ihr AWS Konto zuzugreifen und in Amazon ECR und Amazon EC2 zu schreiben.
- Rolle bereitstellen — Erteilt der Aktion CodeCatalyst Deploy to ECS (im Workflow) die Berechtigung, auf Ihr AWS Konto, Amazon ECS und einige andere AWS Services zuzugreifen.

Weitere Informationen zu IAM-Rollen finden Sie unter [IAM-Rollen](#) im AWS Identity and Access Management Benutzerhandbuch.

### Note

Um Zeit zu sparen, können Sie anstelle der beiden zuvor aufgeführten Rollen eine einzelne `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle, die so genannte Rolle, erstellen. Weitere Informationen finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfangreiche Berechtigungen verfügt, was ein Sicherheitsrisiko darstellen kann. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger wichtig ist. In diesem Tutorial wird davon ausgegangen, dass Sie die beiden zuvor aufgeführten Rollen erstellen.

Um die Build- und Deploy-Rollen zu erstellen, können Sie entweder die AWS Management Console oder die verwenden AWS CLI.

### AWS Management Console


Gehen Sie wie folgt vor, um die Build- und Deploy-Rollen zu erstellen.

Um eine Build-Rolle zu erstellen

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich an bei AWS.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.

- c. Wählen Sie im Navigationsbereich Policies aus.
- d. Wählen Sie Richtlinie erstellen aus.
- e. Wählen Sie den Tab JSON.
- f. Löschen Sie den vorhandenen Code.
- g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

 Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie im Feld Name Folgendes ein:

```
codecatalyst-ecs-build-policy

```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Build-Rolle wie folgt:
  - a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
  - b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie.
  - c. Löschen Sie die bestehende benutzerdefinierte Vertrauensrichtlinie.
  - d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach `codecatalyst-ecs-build-policy` und aktivieren Sie das entsprechende Kontrollkästchen.
- g. Wählen Sie Weiter aus.
- h. Geben Sie als Rollenname Folgendes ein:

**codecatalyst-ecs-build-role**

- i. Geben Sie als Rollenbeschreibung Folgendes ein:

**CodeCatalyst ECS build role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Berechtigungsrichtlinie und einer Vertrauensrichtlinie erstellt.

3. Besorgen Sie sich den ARN für die Build-Rolle wie folgt:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (codecatalyst-ecs-build-role).
  - c. Wählen Sie die Rolle aus der Liste aus.

Die Übersichtsseite der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert. Sie benötigen sie später.

Um eine Bereitstellungsrolle zu erstellen

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich an bei AWS.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
  - c. Wählen Sie im Navigationsbereich Policies aus.
  - d. Wählen Sie Richtlinie erstellen aus.
  - e. Wählen Sie den Tab JSON.
  - f. Löschen Sie den vorhandenen Code.
  - g. Fügen Sie folgenden Code ein:

```
{
  "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

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung. Sie können die Richtlinie dann anhand des Ressourcennamens einschränken, sobald sie verfügbar ist.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie im Feld Name Folgendes ein:

```
codecatalyst-ecs-deploy-policy

```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Bereitstellungsrolle wie folgt:
  - a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
  - b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie.
  - c. Löschen Sie die bestehende benutzerdefinierte Vertrauensrichtlinie.
  - d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {

```

```
        "Service": [
            "codecatalyst-runner.amazonaws.com",
            "codecatalyst.amazonaws.com"
        ],
        "Action": "sts:AssumeRole"
    }
}
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach `codecatalyst-ecs-deploy-policy` und aktivieren Sie das entsprechende Kontrollkästchen.
- g. Wählen Sie Weiter aus.
- h. Geben Sie als Rollenname Folgendes ein:

**codecatalyst-ecs-deploy-role**

- i. Geben Sie als Rollenbeschreibung Folgendes ein:

**CodeCatalyst ECS deploy role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Bereitstellungsrolle mit einer Vertrauensrichtlinie erstellt.

3. Rufen Sie den ARN für die Bereitstellungsrolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-ecs-deploy-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.

Die Übersichtsseite der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert. Sie benötigen sie später.

## AWS CLI

Gehen Sie wie folgt vor, um die Build- und Deploy-Rollen zu erstellen.

## Um eine Vertrauensrichtlinie für beide Rollen zu erstellen

Wie in CodeCatalystECSUser AWS CloudShell, erstellen Sie eine Vertrauensrichtliniendatei:

1. Erstellen Sie die Datei:

```
cat > codecatalyst-ecs-trust-policy.json
```

2. Fügen Sie an der Terminal-Eingabeaufforderung den folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. Platzieren Sie den Cursor hinter der letzten geschweiften Klammer (}).
4. Drücken Sie dann **Enter** und **Ctrl+d**, um die Datei zu speichern und cat zu beenden.

## Um die Build-Richtlinie und die Build-Rolle zu erstellen

1. Erstellen Sie die Build-Richtlinie:

- a. Wie in CodeCatalystECSUser AWS CloudShell, erstellen Sie eine Build-Richtliniendatei:

```
cat > codecatalyst-ecs-build-policy.json
```

- b. Geben Sie an der Eingabeaufforderung den folgenden Code ein:

```
{
```

```

    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "ecr:*",
          "ec2:*"
        ],
        "Resource": "*"
      }
    ]
  }
}

```

- c. Platzieren Sie den Cursor hinter der letzten geschweiften Klammer (}).
  - d. Drücken Sie dann **Enter** und **Ctrl+d**, um die Datei zu speichern und cat zu beenden.
2. Fügen Sie die Build-Richtlinie hinzu zu AWS:

```

aws iam create-policy \
  --policy-name codecatalyst-ecs-build-policy \
  --policy-document file://codecatalyst-ecs-build-policy.json

```

3. Notieren Sie sich in der Befehlsausgabe den "arn": Wert, zum Beispiel `arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy`. Sie benötigen diesen ARN später.
4. Erstellen Sie die Build-Rolle und fügen Sie ihr die Vertrauensrichtlinie hinzu:

```

aws iam create-role \
  --role-name codecatalyst-ecs-build-role \
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json

```

5. Hängen Sie die Build-Richtlinie an die Build-Rolle an:

```

aws iam attach-role-policy \
  --role-name codecatalyst-ecs-build-role \
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy

```

Wobei *arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy* durch den ARN der Build-Richtlinie ersetzt wird, die Sie zuvor notiert haben.

6. Zeigt die Details der Build-Rolle an:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-build-role
```

7. Notieren Sie sich den "Arn": Wert der Rolle, zum Beispiel `arn:aws:iam::111122223333:role/codecatalyst-ecs-build-role`. Sie benötigen diesen ARN später.

Um die Bereitstellungsrichtlinie und die Bereitstellungsrolle zu erstellen

1. Erstellen Sie eine Bereitstellungsrichtlinie:
  - a. Erstellen Sie AWS CloudShell unter eine Bereitstellungsrichtliniendatei:

```
cat > codecatalyst-ecs-deploy-policy.json
```

- b. Geben Sie an der Eingabeaufforderung den folgenden Code ein:

```
{  
  "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

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-

Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"
```

- c. Platzieren Sie den Cursor hinter der letzten geschweiften Klammer (}).
  - d. Drücken Sie dann **Enter** und **Ctrl+d**, um die Datei zu speichern und cat zu beenden.
2. Fügen Sie die Bereitstellungsrichtlinie hinzu zu AWS:

```
aws iam create-policy \  
  --policy-name codecatalyst-ecs-deploy-policy \  
  --policy-document file://codecatalyst-ecs-deploy-policy.json
```

3. Notieren Sie sich in der Befehlsausgabe den "arn": Wert der Bereitstellungsrichtlinie, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` z. B. Sie benötigen diesen ARN später.
4. Erstellen Sie die Bereitstellungsrolle und fügen Sie ihr die Vertrauensrichtlinie hinzu:

```
aws iam create-role \  
  --role-name codecatalyst-ecs-deploy-role \  
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

5. Hängen Sie die Bereitstellungsrichtlinie an die Bereitstellungsrolle an, wobei `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` durch den ARN der Bereitstellungsrichtlinie ersetzt wird, die Sie zuvor notiert haben.

```
aws iam attach-role-policy \  
  --role-name codecatalyst-ecs-deploy-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy
```

6. Zeigen Sie die Details der Bereitstellungsrolle an:

```
aws iam get-role \  
  --role-name codecatalyst-ecs-deploy-role
```

7. Notieren Sie sich den "Arn": Wert der Rolle, zum Beispiel `arn:aws:iam::111122223333:role/codecatalyst-ecs-deploy-role`. Sie benötigen diesen ARN später.

## Schritt 5: AWS Rollen hinzufügen CodeCatalyst

In diesem Schritt fügen Sie der CodeCatalyst Kontoverbindung in Ihrem Bereich die Build-Rolle (`codecatalyst-ecs-build-role`) und die Bereitstellungsrolle (`codecatalyst-ecs-deploy-role`) hinzu.

Um deiner Kontoverbindung Rollen zum Erstellen und Bereitstellen hinzuzufügen

1. Navigieren CodeCatalyst Sie darin zu Ihrem Bereich.
2. Wählen Sie AWS accounts (-Konten). Eine Liste der Kontoverbindungen wird angezeigt.
3. Wählen Sie die Kontoverbindung aus, die dem AWS Konto entspricht, in dem Sie Ihre Build- und Deploy-Rollen erstellt haben.
4. Wählen Sie in der AWS Managementkonsole die Option Rollen verwalten aus.

Die Seite „IAM-Rolle zu Amazon CodeCatalyst Space hinzufügen“ wird angezeigt. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

5. Wählen Sie Eine bestehende Rolle hinzufügen, die Sie in IAM erstellt haben.

Eine Dropdownliste wird angezeigt. In der Liste werden alle IAM-Rollen mit einer Vertrauensrichtlinie angezeigt, die die Dienstprinzipale `codecatalyst-runner.amazonaws.com` und die `codecatalyst.amazonaws.com` Dienstprinzipale umfasst.

6. Wählen Sie `codecatalyst-ecs-build-role` in der Dropdownliste die Option Rolle hinzufügen aus.

### Note

Wenn Sie das sehen `The security token included in the request is invalid`, liegt es möglicherweise daran, dass Sie nicht über die richtigen Berechtigungen verfügen. Um dieses Problem zu beheben, melden Sie sich ab und melden Sie sich mit dem AWS Konto wieder an, das Sie bei der Erstellung Ihres CodeCatalyst Bereichs verwendet haben. AWS

7. Wählen Sie „IAM-Rolle hinzufügen“, wählen Sie „Eine bestehende Rolle hinzufügen, die Sie in IAM erstellt haben“ und wählen Sie in der Drop-down-Liste die Option aus. `codecatalyst-ecs-deploy-role` Wählen Sie Rolle hinzufügen aus.

Sie haben jetzt die Build- und Deploy-Rollen zu Ihrem Bereich hinzugefügt.



8. Kopieren Sie den Wert des CodeCatalyst Amazon-Anzeigenamens. Sie benötigen diesen Wert später, wenn Sie Ihren Workflow erstellen.

## Schritt 6: Erstellen Sie ein Quell-Repository

In diesem Schritt erstellen Sie ein Quell-Repository in CodeCatalyst. In diesem Repository werden die Quelldateien des Tutorials gespeichert, z. B. die Aufgabendefinitionsdatei.

Weitere Informationen zu Quell-Repositorys finden Sie unter [Erstellen eines Quell-Repositorys](#).

Um ein Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt, `codecatalyst-ecs-project`.
3. Wählen Sie im Navigationsbereich Code und dann Source Repositories aus.
4. Wählen Sie Repository hinzufügen und anschließend Repository erstellen aus.
5. Geben Sie im Feld Repository-Name Folgendes ein:

```
codecatalyst-ecs-source-repository
```

6. Wählen Sie Erstellen.

## Schritt 7: Quelldateien hinzufügen

In diesem Abschnitt fügen Sie die Hello World-Quelldateien zu Ihrem CodeCatalyst Repository hinzu. `codecatalyst-ecs-source-repository` Sie bestehen aus:

- Eine `index.html` Datei — Zeigt eine Hello World-Nachricht im Browser an.
- Ein Dockerfile — Beschreibt das Basis-Image, das für Ihr Docker-Image verwendet werden soll, und die Docker-Befehle, die darauf angewendet werden sollen.
- Eine `taskdef.json` Datei — Definiert das Docker-Image, das beim Starten von Aufgaben in Ihrem Cluster verwendet werden soll.

Die Ordnerstruktur ist wie folgt:

```
.
```

```
|– public-html
|   |– index.html
|– Dockerfile
|– taskdef.json
```

### Note

Die folgenden Anweisungen zeigen Ihnen, wie Sie die Dateien mithilfe der CodeCatalyst Konsole hinzufügen. Sie können jedoch auch Git verwenden, wenn Sie dies bevorzugen. Details hierzu finden Sie unter [Klonen eines Quell-Repositorys](#).

## Themen

- [index.html](#)
- [Dockerfile](#)
- [taskdef.json](#)

## index.html

Die `index.html` Datei zeigt im Browser eine Hello World-Nachricht an.

Um die Datei `index.html` hinzuzufügen

1. Gehen Sie in der CodeCatalyst Konsole zu Ihrem Quell-Repository, `codecatalyst-ecs-source-repository`.
2. Wählen Sie unter Dateien die Option Datei erstellen aus.
3. Geben Sie als Dateiname Folgendes ein:

```
public-html/index.html
```

### Important

Stellen Sie sicher, dass Sie das `public-html/` Präfix angeben, um einen Ordner mit demselben Namen zu erstellen. Das `index.html` wird sich voraussichtlich in diesem Ordner befinden.

4. Geben Sie in das Textfeld den folgenden Code ein:

```
<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. Wählen Sie Commit und anschließend erneut Commit aus.

Das `index.html` wird Ihrem Repository in einem `public-html` Ordner hinzugefügt.

## Dockerfile

Das Dockerfile beschreibt das zu verwendende Basis-Decker-Image und die darauf anzuwendenden Docker-Befehle. [Weitere Informationen zum Dockerfile finden Sie in der Dockerfile-Referenz.](#)

Das hier angegebene Dockerfile gibt an, dass das Apache 2.4-Basisimage (`httpd`) verwendet werden soll. Es enthält auch Anweisungen zum Kopieren einer `index.html` aufgerufenen Quelldatei in einen Ordner auf dem Apache-Server, der Webseiten bereitstellt. Die `EXPOSE` Anweisung in der Dockerfile teilt Docker mit, dass der Container auf Port 80 lauscht.

Um das Dockerfile hinzuzufügen

1. Wählen Sie in Ihrem Quell-Repository die Option Datei erstellen.
2. Geben Sie als Dateiname Folgendes ein:

**Dockerfile**

Geben Sie keine Dateierweiterung an.

**⚠ Important**

Das Dockerfile muss sich im Stammordner Ihres Repositorys befinden. Der Docker `build` Befehl des Workflows erwartet, dass es dort vorhanden ist.

3. Geben Sie in das Textfeld den folgenden Code ein:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

4. Wählen Sie Commit und anschließend erneut Commit aus.

Das Dockerfile wird Ihrem Repository hinzugefügt.

**taskdef.json**

Die `taskdef.json` Datei, die Sie in diesem Schritt hinzufügen, ist dieselbe wie die, in der Sie bereits angegeben haben, [Schritt 2: Stellen Sie eine Platzhalteranwendung in Amazon ECS bereit](#) mit dem folgenden Unterschied:

Anstatt einen fest codierten Docker-Imagename im `image:` Feld (`httpd:2.4`) anzugeben, verwendet die Aufgabendefinition hier einige Variablen, um das Bild zu bezeichnen: `und`. `$REPOSITORY_URI` `$IMAGE_TAG` Diese Variablen werden durch echte Werte ersetzt, die durch die Build-Aktion des Workflows generiert wurden, wenn Sie den Workflow in einem späteren Schritt ausführen.

Einzelheiten zu den Aufgabendefinitionsparametern finden Sie unter [Aufgabendefinitionsparameter](#) im Amazon Elastic Container Service Developer Guide.

Um die Datei `taskdef.json` hinzuzufügen

1. Wählen Sie in Ihrem Quell-Repository die Option Datei erstellen aus.
2. Geben Sie als Dateiname Folgendes ein:

```
taskdef.json
```

3. Geben Sie in das Textfeld den folgenden Code ein:

```
{
  "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"
}
```

Ersetzen Sie im vorherigen Code

*arn:aws:iam: :account\_id:role/ -role codecatalyst-ecs-task-execution*

mit dem ARN der Aufgabenausführungsrolle, die Sie notiert haben [Um die Rolle zur Aufgabenausführung zu erstellen](#).

4. Wählen Sie Commit und anschließend erneut Commit aus.

Die taskdef.json Datei wird Ihrem Repository hinzugefügt.

## Schritt 8: Einen Workflow erstellen und ausführen

In diesem Schritt erstellen Sie einen Workflow, der Ihre Quelldateien zu einem Docker-Image zusammenbaut und das Image dann in Ihrem Amazon ECS-Cluster bereitstellt. Diese Bereitstellung ersetzt die bestehende Apache-Platzhalteranwendung.

Der Workflow besteht aus den folgenden Bausteinen, die nacheinander ausgeführt werden:

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine Build-Aktion (`BuildBackend`) — Beim Auslösen erstellt die Aktion das Docker-Image mithilfe der Dockerfile und überträgt das Image an Amazon ECR. Die Build-Aktion aktualisiert auch das `taskdef.json` mit dem richtigen `image` Feldwert und erstellt dann ein Ausgabeartefakt dieser Datei. Dieses Artefakt wird als Eingabe für die Bereitstellungsaktion verwendet, die als Nächstes folgt.

Weitere Informationen zur Build-Aktion finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#).

- Eine Bereitstellungsaktion (`DeployToECS`) — Nach Abschluss der Build-Aktion sucht die Bereitstellungsaktion nach dem von der Build-Aktion (`TaskDefArtifact`) generierten Ausgabeartefakt, findet dessen Inhalt und registriert es bei Ihrem Amazon ECS-Service. `taskdef.json` Der Service folgt dann den Anweisungen in der `taskdef.json` Datei, um drei Amazon ECS-Aufgaben — und zugehörige Hello World Docker-Container — in Ihrem Amazon ECS-Cluster auszuführen.

So erstellen Sie ein Workflow

1. Wählen Sie in der CodeCatalyst Konsole im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie Workflow erstellen aus.
3. Wählen Sie für Quell-Repository die Option `codecatalyst-ecs-source-repository`.
4. Wählen Sie für Branch die Option `main`.
5. Wählen Sie Erstellen.
6. Löschen Sie den YAML-Beispielcode.
7. Fügen Sie den folgenden YAML-Code hinzu:

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
```

Ersetzen Sie im vorherigen Code:

- Beide Instanzen von *codecatalyst-ecs-environment* mit dem Namen der Umgebung, in der Sie erstellt haben [Voraussetzungen](#).
- Beide Instanzen von *codecatalyst-account-connection* mit dem Anzeigenamen Ihrer Kontoverbindung. Der Anzeigename kann eine Zahl sein. Weitere Informationen finden Sie unter [Schritt 5: AWS Rollen hinzufügen CodeCatalyst](#).
- *codecatalyst-ecs-build-role* mit dem Namen der Build-Rolle, in der Sie sie erstellt haben [Schritt 4: AWS Rollen erstellen](#).



- `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo` (in der `Value`: Eigenschaft) mit der URI des Amazon ECR-Repositorys, in dem Sie es erstellt haben. [Schritt 3: Erstellen Sie ein Amazon ECR-Image-Repository](#)
- `111122223333.dkr.ecr.us-west-2.amazonaws.com` (im `Run`: `aws ecr` Befehl) mit der URI des Amazon ECR-Repositorys ohne das Bildsuffix (`.`). `/codecatalyst-ecs-image-repo`
- `codecatalyst-ecs-deploy-role` mit dem Namen der Bereitstellungsrolle, in der Sie sie erstellt haben. [Schritt 4: AWS Rollen erstellen](#)
- Beide Instanzen von `us-west-2` mit Ihrem AWS Regionalcode. Eine Liste der Regionalcodes finden Sie unter [Regionale Endpunkte](#) in der Allgemeinen AWS-Referenz

#### Note

Wenn Sie sich entschieden haben, keine Build- und Deploy-Rollen zu erstellen, ersetzen Sie `codecatalyst-ecs-build-role` und `codecatalyst-ecs-deploy-role` durch den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle. Weitere Informationen über diese Rolle finden Sie unter [Schritt 4: AWS Rollen erstellen](#).

#### Tip

Anstatt die `sed` Befehle `find` und `und` zu verwenden, die im vorherigen Workflow-Code gezeigt wurden, um das Repository und den Namen des Images zu aktualisieren, können Sie zu diesem Zweck die Amazon ECS-Aufgabendefinitionsaktion `render` verwenden. Weitere Informationen finden Sie unter [Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen](#).

8. (Optional) Wählen Sie „Validieren“, um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie ihn festschreiben.
9. Wählen Sie Commit (Übergeben).
10. Geben Sie im Dialogfeld „Workflow bestätigen“ Folgendes ein:
  - a. Entfernen Sie bei Nachricht bestätigen den Text und geben Sie Folgendes ein:

**Add first workflow**

- b. Wählen Sie für `Repositorycodecatalyst-ecs-source-repository`.
- c. Wählen Sie als Branch-Name die Option `main` aus.
- d. Wählen Sie `Commit` (Übergeben).

Sie haben jetzt einen Workflow erstellt. Eine Workflow-Ausführung wird aufgrund des oben im Workflow definierten Triggers automatisch gestartet. Insbesondere, als Sie die `workflow.yaml` Datei in Ihr Quell-Repository übernommen (und per Push übertragen) haben, hat der Trigger die Workflow-Ausführung gestartet.

Um den Fortschritt der Workflow-Ausführung zu sehen

1. Wählen Sie im Navigationsbereich der CodeCatalyst Konsole CI/CD und dann Workflows aus.
2. Wählen Sie den Workflow aus, den Sie gerade erstellt haben, `codecatalyst-ecs-workflow`.
3. Wählen Sie `BuildBackend`, ob Sie den Baufortschritt sehen möchten.
4. Wählen Sie `DeployToECS`, um den Fortschritt der Bereitstellung zu sehen.

Weitere Informationen zum Anzeigen von Ausführungsdetails finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).

Um die Bereitstellung zu überprüfen

1. Öffnen Sie die klassische Amazon-ECS-Konsole unter <https://console.aws.amazon.com/ecs/>.
2. Wählen Sie Ihren Cluster, `codecatalyst-ecs-cluster`.
3. Wählen Sie die Registerkarte `Tasks` aus.
4. Wählen Sie eine der drei Aufgaben.
5. Wählen Sie im Feld `Öffentliche IP` die Option `Open Address` aus.

Im Browser wird eine Seite „Hello World“ angezeigt, die darauf hinweist, dass der Amazon ECS-Service Ihre Anwendung erfolgreich bereitgestellt hat.

## Schritt 9: Nehmen Sie eine Änderung an Ihren Quelldateien vor

In diesem Abschnitt nehmen Sie eine Änderung an der `index.html` Datei in Ihrem Quell-Repository vor. Diese Änderung veranlasst den Workflow, ein neues Docker-Image zu erstellen, es mit

einer Commit-ID zu kennzeichnen, es an Amazon ECR weiterzuleiten und es in Amazon ECS bereitzustellen.

Um die Datei `index.html` zu ändern

1. Wählen Sie in der CodeCatalyst Konsole im Navigationsbereich Code, dann Quell-Repositories und anschließend Ihr Repository aus. `codecatalyst-ecs-source-repository`
2. Klicken Sie auf `public-html` und danach auf `index.html`.

Der Inhalt von `index.html` wird angezeigt.

3. Wählen Sie Bearbeiten aus.
4. Ändern Sie in Zeile 14 den Hello World Text in `Tutorial complete!`.
5. Wählen Sie Commit und anschließend erneut Commit aus.

Durch den Commit wird ein neuer Workflow-Lauf gestartet.

6. (Optional) Gehen Sie zur Hauptseite Ihres Quell-Repositorys, wählen Sie Commits anzeigen und notieren Sie sich die Commit-ID für die `index.html` Änderung.
7. Beobachten Sie den Fortschritt der Bereitstellung:
  - a. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  - b. Wählen Sie `codecatalyst-ecs-workflow`, ob Sie die letzte Ausführung anzeigen möchten.
  - c. Wählen Sie `DeployToECS BuildBackend`, um den Fortschritt der Workflow-Ausführung zu sehen.
8. Stellen Sie wie folgt sicher, dass Ihre Anwendung aktualisiert wurde:
  - a. Öffnen Sie die klassische Amazon-ECS-Konsole unter <https://console.aws.amazon.com/ecs/>.
  - b. Wählen Sie Ihren Cluster, `codecatalyst-ecs-cluster`.
  - c. Wählen Sie die Registerkarte Tasks aus.
  - d. Wählen Sie eine der drei Aufgaben.
  - e. Wählen Sie im Feld Öffentliche IP die Option Open Address aus.

Eine `Tutorial complete!` Seite wird angezeigt.

9. (Optional) Wechseln Sie in AWS zur Amazon ECR-Konsole und überprüfen Sie, ob das neue Docker-Image mit der Commit-ID aus Schritt 6 gekennzeichnet wurde.

## Bereinigen

Bereinigen Sie die in diesem Tutorial verwendeten Dateien und Dienste, um zu vermeiden, dass dafür Gebühren anfallen.

Im AWS Management Console, bereinigen Sie in dieser Reihenfolge:

1. Gehen Sie in Amazon ECS wie folgt vor:
  - a. Löschencodecatalyst-ecs-service.
  - b. Löschencodecatalyst-ecs-cluster.
  - c. Abmeldencodecatalyst-ecs-task-definition.
2. Löschen codecatalyst-ecs-image-repo Sie in Amazon ECR.
3. Löschen codecatalyst-ecs-security-group Sie in Amazon EC2.
4. Löschen Sie im IAM Identity Center:
  - a. CodeCatalystECSUser
  - b. CodeCatalystECSPermissionSet

Bereinigen Sie in der CodeCatalyst Konsole wie folgt:


1. Löschencodecatalyst-ecs-workflow.
2. Löschencodecatalyst-ecs-environment.
3. Löschencodecatalyst-ecs-source-repository.
4. Löschencodecatalyst-ecs-project.

In diesem Tutorial haben Sie gelernt, wie Sie mithilfe eines CodeCatalyst Workflows und einer Aktion „In Amazon ECS bereitstellen“ eine Anwendung für einen Amazon ECS-Service bereitstellen.

## Tutorial: Bereitstellen einer Anwendung in Amazon EKS

In diesem Tutorial erfahren Sie, wie Sie mithilfe eines CodeCatalyst Amazon-Workflows, Amazon EKS und einiger anderer Services eine containerisierte Anwendung in Amazon Elastic Kubernetes Service bereitstellen. AWS Bei der bereitgestellten Anwendung handelt es sich um ein einfaches „Hello, World!“ Website, die auf einem Docker-Image des Apache-Webserver basiert. Das Tutorial führt Sie durch die erforderlichen Vorbereitungsarbeiten wie das Einrichten einer Entwicklungsmaschine und eines Amazon EKS-Clusters und beschreibt anschließend, wie Sie einen Workflow erstellen, um die Anwendung zu erstellen und sie im Cluster bereitzustellen.

Nach Abschluss der ersten Bereitstellung werden Sie im Tutorial angewiesen, eine Änderung an Ihrer Anwendungsquelle vorzunehmen. Diese Änderung führt dazu, dass ein neues Docker-Image erstellt und mit neuen Revisionsinformationen in Ihr Docker-Image-Repository übertragen wird. Die neue Version des Docker-Images wird dann in Amazon EKS bereitgestellt.

 Tip

Anstatt sich durch dieses Tutorial zu arbeiten, können Sie einen Blueprint verwenden, der ein vollständiges Amazon EKS-Setup für Sie durchführt. Sie müssen den EKS App Deployment Blueprint verwenden. Weitere Informationen finden Sie unter [Ein Projekt mit einem Blueprint erstellen](#).

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Richten Sie Ihren Entwicklungscomputer ein](#)
- [Schritt 2: Erstellen Sie einen Amazon EKS-Cluster](#)
- [Schritt 3: Erstellen Sie ein Amazon ECR-Image-Repository](#)
- [Schritt 4: Quelldateien hinzufügen](#)
- [Schritt 5: AWS Rollen erstellen](#)
- [Schritt 6: AWS Rollen hinzufügen CodeCatalyst](#)
- [Schritt 7: Aktualisieren Sie das ConfigMap](#)
- [Schritt 8: Erstellen Sie einen Workflow und führen Sie ihn aus](#)
- [Schritt 9: Nehmen Sie eine Änderung an Ihren Quelldateien vor](#)
- [Bereinigen](#)

## Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen:

- Sie benötigen einen CodeCatalyst Amazon-Bereich mit einem verbundenen AWS Konto. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).
- In Ihrem Bereich benötigen Sie ein leeres CodeCatalyst Projekt mit dem Namen „Von vorne beginnen“:

```
codecatalyst-eks-project
```

Weitere Informationen finden Sie unter [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#).

- In Ihrem Projekt benötigen Sie ein leeres CodeCatalyst Quell-Repository mit dem Namen:

```
codecatalyst-eks-source-repository
```

Weitere Informationen finden Sie unter [Quell-Repositoryn in CodeCatalyst](#).

- In Ihrem Projekt benötigen Sie eine CodeCatalyst CI/CD-Umgebung (keine Entwicklungsumgebung) namens:

```
codecatalyst-eks-environment
```

Konfigurieren Sie diese Umgebung wie folgt:

- Wählen Sie einen beliebigen Typ aus, z. B. „Keine Produktion“.
- Connect dein AWS Konto damit.

Weitere Informationen finden Sie unter [Arbeiten mit Umgebungen](#).

## Schritt 1: Richten Sie Ihren Entwicklungscomputer ein

Der erste Schritt in diesem Tutorial besteht darin, einen Entwicklungscomputer mit einigen Tools zu konfigurieren, die Sie in diesem Tutorial verwenden werden. Diese Tools sind:

- das `eksctl` Hilfsprogramm — für die Cluster-Erstellung
- das `kubectl` Hilfsprogramm — eine Voraussetzung für `eksctl`
- das AWS CLI — auch eine Voraussetzung für `eksctl`

Sie können diese Tools auf Ihrem vorhandenen Entwicklungscomputer installieren, falls Sie einen haben, oder Sie können eine CodeCatalyst Entwicklungsumgebung verwenden, die cloudbasiert ist. Der Vorteil einer CodeCatalyst Entwicklungsumgebung besteht darin, dass sie einfach hoch- und heruntergefahren werden kann und in andere CodeCatalyst Dienste integriert ist, sodass Sie dieses Tutorial in weniger Schritten durcharbeiten können.

In diesem Tutorial wird davon ausgegangen, dass Sie eine CodeCatalyst Entwicklungsumgebung verwenden.

Die folgenden Anweisungen beschreiben eine schnelle Methode, um eine CodeCatalyst Entwicklungsumgebung zu starten und sie mit den erforderlichen Tools zu konfigurieren. Genauere Anweisungen finden Sie unter:

- [Erstellen einer Entwicklungsumgebung](#) in dieser Anleitung.
- [Installation von kubectl](#) im Amazon EKS-Benutzerhandbuch.
- [Installation oder Aktualisierung von eksctl](#) im Amazon EKS-Benutzerhandbuch.
- [Installation oder Aktualisierung der neuesten Version von, die AWS CLI](#) im AWS Command Line Interface Benutzerhandbuch beschrieben ist.

Um eine Entwicklungsumgebung zu starten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt, `codecatalyst-eks-project`.
3. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.
4. Wählen Sie den Namen Ihres Quell-Repositorys, `codecatalyst-eks-source-repository`.
5. Wählen Sie oben Create Dev Environment und dann AWS Cloud9 (im Browser) aus.
6. Vergewissern Sie sich, dass In vorhandenem Zweig und Hauptbereich arbeiten ausgewählt sind, und wählen Sie dann Create aus.

Ihre Entwicklungsumgebung wird in einem neuen Browser-Tab gestartet, in den Ihr Repository (`codecatalyst-eks-source-repository`) geklont wird.

Um kubectl zu installieren und zu konfigurieren

1. Geben Sie im Dev Environment-Terminal Folgendes ein:

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
```

2. Geben Sie ein:

```
chmod +x ./kubectl
```

### 3. Geben Sie ein:

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
```

### 4. Geben Sie ein:

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

### 5. Geben Sie ein:

```
kubectl version --short --client
```

### 6. Vergewissern Sie sich, dass eine Version angezeigt wird.

Sie haben es jetzt installiert `kubectl`.

Um `eksctl` zu installieren und zu konfigurieren

#### Note

`eksctl` ist nicht unbedingt erforderlich, da Sie stattdessen verwenden `kubectl` können. `eksctl` hat jedoch den Vorteil, dass ein Großteil der Clusterkonfiguration automatisiert wird, und ist daher das für dieses Tutorial empfohlene Tool.

### 1. Geben Sie im Dev Environment-Terminal Folgendes ein:

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
```

### 2. Geben Sie ein:

```
sudo cp /tmp/eksctl /usr/bin
```

### 3. Geben Sie ein:

```
eksctl version
```

### 4. Vergewissern Sie sich, dass eine Version angezeigt wird.

Sie haben es jetzt installiert `eksctl`.



Um zu überprüfen, ob das installiert AWS CLI ist

1. Geben Sie im Dev Environment-Terminal Folgendes ein:

```
aws --version
```

2. Überprüfen Sie, ob eine Version angezeigt wird, um zu überprüfen, ob die installiert AWS CLI ist.


Führen Sie die verbleibenden Verfahren aus, um die AWS CLI mit den erforderlichen Zugriffsberechtigungen zu konfigurieren AWS.

Um das zu konfigurieren AWS CLI

Sie müssen das AWS CLI mit Zugriffsschlüsseln und einem Sitzungstoken konfigurieren, um ihm Zugriff auf AWS Dienste zu gewähren. Die folgenden Anweisungen bieten eine schnelle Möglichkeit, die Schlüssel und das Token zu konfigurieren. Wenn Sie jedoch detaillierte Anweisungen wünschen, finden Sie [AWS CLI im AWS Command Line Interface Benutzerhandbuch unter Konfiguration von](#).

1. Erstellen Sie wie folgt einen IAM Identity Center-Benutzer:
  - a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS IAM Identity Center Konsole unter <https://console.aws.amazon.com/singlesignon/>.

(Möglicherweise müssen Sie Aktivieren auswählen, wenn Sie sich noch nie bei IAM Identity Center angemeldet haben.)

 Note

Stellen Sie sicher, dass Sie sich mit dem anmelden AWS-Konto , der mit Ihrem CodeCatalyst Bereich verbunden ist. Sie können überprüfen, welches Konto verbunden ist, indem Sie zu Ihrem Bereich navigieren und den Tab AWS-Konten auswählen. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).

- b. Wählen Sie im Navigationsbereich Users (Benutzer) und dann Add User (Benutzer hinzufügen) aus.
- c. Geben Sie im Feld Nutzernamen Folgendes ein:

```
codecatalyst-eks-user
```

- d. Wählen Sie unter Passwort die Option Einmalpasswort generieren aus, das Sie mit diesem Benutzer teilen können.
- e. Geben Sie in den Feldern E-Mail-Adresse und E-Mail-Adresse bestätigen eine E-Mail-Adresse ein, die noch nicht in IAM Identity Center existiert.
- f. Geben Sie im Feld Vorname Folgendes ein:

**codecatalyst-eks-user**

- g. Geben Sie im Feld Nachname Folgendes ein:

**codecatalyst-eks-user**

- h. Behalten Sie im Feld Anzeigename Folgendes bei:

**codecatalyst-eks-user codecatalyst-eks-user**

- i. Wählen Sie Weiter aus.
- j. Wählen Sie auf der Seite Benutzer zu Gruppen hinzufügen die Option Weiter aus.
- k. Überprüfen Sie auf der Seite Benutzer überprüfen und hinzufügen die Informationen und wählen Sie Benutzer hinzufügen aus.

Ein Dialogfeld mit einem Einmalkennwort wird angezeigt.


- l. Wählen Sie Kopieren und fügen Sie dann die Anmeldeinformationen in eine Textdatei ein. Die Anmeldeinformationen bestehen aus der URL des AWS Zugriffsportals, einem Benutzernamen und einem Einmalkennwort.
  - m. Klicken Sie auf Schließen.
2. Erstellen Sie wie folgt einen Berechtigungssatz:
    - a. Wählen Sie im Navigationsbereich die Option Berechtigungssätze und dann Berechtigungssatz erstellen aus.
    - b. Wählen Sie Vordefinierter Berechtigungssatz und dann aus AdministratorAccess. Diese Richtlinie gewährt allen volle Berechtigungen AWS-Services.
    - c. Wählen Sie Weiter aus.
    - d. Entfernen Sie im Feld Name des Berechtigungssatzes den Wert AdministratorAccess und geben Sie Folgendes ein:

`codecatalyst-eks-permission-set`

- e. Wählen Sie Weiter aus.
  - f. Überprüfen Sie auf der Seite Überprüfen und erstellen die Informationen und wählen Sie Erstellen aus.
3. Weisen Sie den Berechtigungssatz wie folgt zu `codecatalyst-eks-user`:
- a. Wählen Sie im Navigationsbereich das Kontrollkästchen neben dem aus AWS-Konten, bei dem Sie derzeit angemeldet sind AWS-Konto , und aktivieren Sie es anschließend.
  - b. Wählen Sie Benutzer oder Gruppen zuweisen aus.
  - c. Wählen Sie die Registerkarte Users.
  - d. Aktivieren Sie das Kontrollkästchen neben `codecatalyst-eks-user`.
  - e. Wählen Sie Weiter aus.
  - f. Aktivieren Sie das Kontrollkästchen neben `codecatalyst-eks-permission-set`.
  - g. Wählen Sie Weiter aus.
  - h. Überprüfen Sie die Informationen und wählen Sie Senden aus.

Sie haben sie nun `codecatalyst-eks-permission-set` zugewiesen `codecatalyst-eks-user` und an Sie AWS-Konto gebunden.

4. Rufen Sie `codecatalyst-eks-user` die Zugriffsschlüssel und das Sitzungstoken wie folgt ab:
- a. Stellen Sie sicher, dass Sie die URL des AWS Zugriffsportals sowie den Benutzernamen und das Einmalpasswort für `habencodecatalyst-eks-user`. Sie hätten diese Informationen früher in einen Texteditor kopieren sollen.

 Note

Wenn Ihnen diese Informationen nicht vorliegen, rufen Sie die `codecatalyst-eks-user` Detailseite im IAM Identity Center auf und wählen Sie Passwort zurücksetzen, Einmalpasswort generieren [...], und klicken Sie erneut auf Passwort zurücksetzen, um die Informationen auf dem Bildschirm anzuzeigen.

- b. Melden Sie sich ab AWS.
- c. Fügen Sie die URL des AWS Zugangsportals in die Adressleiste Ihres Browsers ein.
- d. Melden Sie sich an mit:

- **Nutzername:**

```
codecatalyst-eks-user
```

- **Passwort:**

*one-time-password*

- e. Geben Sie unter Neues Passwort einrichten ein neues Passwort ein und wählen Sie Neues Passwort festlegen aus.

Auf dem Bildschirm erscheint ein AWS-KontoFeld.


- f. Wählen Sie AWS-Kontound wählen Sie dann den Namen des Benutzers und des AWS-Konto Berechtigungssatzes aus, dem Sie den `codecatalyst-eks-user` Benutzer zugewiesen haben.
- g. Wählen Sie neben `codecatalyst-eks-permission-set` Befehlszeile oder programmatischer Zugriff aus.
- h. Kopieren Sie die Befehle in der Mitte der Seite. Sie sehen etwa wie folgt aus:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... wobei *Session-Token* eine lange zufällige Zeichenfolge ist.

5. Fügen Sie die Zugriffsschlüssel und das Sitzungstoken wie AWS CLI folgt hinzu:
  - a. Kehren Sie zu Ihrer CodeCatalyst Entwicklungsumgebung zurück.
  - b. Fügen Sie an der Terminal-Eingabeaufforderung die Befehle ein, die Sie kopiert haben. Drücken Sie die Eingabetaste.

Sie haben das jetzt AWS CLI mit Zugriffsschlüsseln und einem Sitzungstoken konfiguriert. Sie können es jetzt verwenden AWS CLI , um die für dieses Tutorial erforderlichen Aufgaben zu erledigen.

 **Important**

Wenn Sie während dieses Tutorials zu irgendeinem Zeitpunkt Meldungen sehen, die den folgenden ähneln:

Unable to locate credentials. You can configure credentials by running "aws configure".

Oder:

ExpiredToken: The security token included in the request is expired

... das liegt daran, dass Ihre AWS CLI Sitzung abgelaufen ist. Führen Sie in diesem Fall den `aws configure` Befehl nicht aus. Verwenden Sie stattdessen die Anweisungen in Schritt 4 dieses Verfahrens, das mit „Aktualisieren Obtain codecatalyst-eks-user's access key and session token Sie Ihre Sitzung“ beginnt.

## Schritt 2: Erstellen Sie einen Amazon EKS-Cluster

In diesem Abschnitt erstellen Sie einen Cluster in Amazon EKS. Die folgenden Anweisungen beschreiben eine schnelle Methode zum Erstellen des Clusters mithilfe von `eksctl`. Wenn Sie jedoch detaillierte Anweisungen wünschen, finden Sie unter:

- [Erste Schritte mit eksctl](#) im Amazon EKS-Benutzerhandbuch
- or
- [Erste Schritte mit der Konsole und AWS CLI](#) im Amazon EKS-Benutzerhandbuch (dieses Thema enthält `kubectl` Anweisungen zum Erstellen des Clusters)

### Note

[Private Cluster](#) werden von der CodeCatalyst Integration mit Amazon EKS nicht unterstützt.

Bevor Sie beginnen

Stellen Sie sicher, dass Sie die folgenden Aufgaben auf Ihrem Entwicklungscomputer abgeschlossen haben:

- Das `eksctl` Hilfsprogramm wurde installiert.
- Das `kubectl` Hilfsprogramm wurde installiert.
- Habe das installiert AWS CLI und mit Zugriffsschlüsseln und einem Sitzungstoken konfiguriert.

Informationen zur Ausführung dieser Aufgaben finden Sie unter [Schritt 1: Richten Sie Ihren Entwicklungscomputer ein](#).

So erstellen Sie einen Cluster

 **Important**

Verwenden Sie nicht die Benutzeroberfläche des Amazon EKS-Service, um den Cluster zu erstellen, da der Cluster dann nicht korrekt konfiguriert wird. Verwenden Sie das `eksctl` Hilfsprogramm, wie in den folgenden Schritten beschrieben.

1. Gehen Sie zu Ihrer Entwicklungsumgebung.
2. Erstellen Sie einen Cluster und Knoten:


```
eksctl create cluster --name codecatalyst-eks-cluster --region us-west-2
```

Wobei gilt:

- *codecatalyst-eks-cluster* wird durch den Namen ersetzt, den Sie Ihrem Cluster geben möchten.
- *us-west-2* wird durch Ihre Region ersetzt.

Nach 10 bis 20 Minuten wird eine Meldung ähnlich der folgenden angezeigt:

```
EKS cluster "codecatalyst-eks-cluster" in "us-west-2" region is ready
```

 **Note**

Während AWS der Erstellung Ihres Clusters werden mehrere `waiting for CloudFormation stack` Meldungen angezeigt. Das ist normal.

3. Stellen Sie sicher, dass Ihr Cluster erfolgreich erstellt wurde:

```
kubectl cluster-info
```

Es wird eine Meldung ähnlich der folgenden angezeigt, die auf eine erfolgreiche Clustererstellung hinweist:

```
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
```

### Schritt 3: Erstellen Sie ein Amazon ECR-Image-Repository

In diesem Abschnitt erstellen Sie ein privates Image-Repository in Amazon Elastic Container Registry (Amazon ECR). Dieses Repository speichert das Docker-Image für das Tutorial.

Weitere Informationen zu Amazon ECR finden Sie im Amazon Elastic Container Registry User Guide.

Um ein Bild-Repository in Amazon ECR zu erstellen

1. Gehen Sie zu Ihrer Entwicklungsumgebung.
2. Erstellen Sie ein leeres Repository in Amazon ECR:

```
aws ecr create-repository --repository-name codecatalyst-eks-image-repo
```

*codecatalyst-eks-image-repo* Ersetzen Sie es durch den Namen, den Sie dem Amazon ECR-Repository geben möchten.

In diesem Tutorial wird davon ausgegangen, dass Sie Ihrem Repository *codecatalyst-eks-image-repo* einen Namen gegeben haben.

3. Zeigen Sie die Details des Amazon ECR-Repositorys an:

```
aws ecr describe-repositories \  
  --repository-names codecatalyst-eks-image-repo
```

4. Notieren Sie sich den "repositoryUri"-Wert, zum Beispiel `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo`

Sie benötigen ihn später, wenn Sie das Repository zu Ihrem Workflow hinzufügen.

### Schritt 4: Quelldateien hinzufügen

In diesem Abschnitt fügen Sie Anwendungsquelldateien zu Ihrem Quell-Repository hinzu (*codecatalyst-eks-source-repository*). Sie bestehen aus:

- Eine `index.html` Datei — Zeigt ein „Hallo, Welt!“ Nachricht im Browser.
- Ein Dockerfile — Beschreibt das Basis-Image, das für Ihr Docker-Image verwendet werden soll, und die Docker-Befehle, die darauf angewendet werden sollen.
- Eine `deployment.yaml` Datei — Das Kubernetes-Manifest, das den Kubernetes-Service und die Bereitstellung definiert.

Die Ordnerstruktur sieht wie folgt aus:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
  |   |– index.html
  |– Dockerfile
```

## Themen

- [index.html](#)
- [Dockerfile](#)
- [deployment.yaml](#)

## index.html

In der `index.html` Datei wird die Meldung „Hello, World!“ angezeigt Nachricht im Browser.

Um die Datei `index.html` hinzuzufügen

1. Gehen Sie zu Ihrer Entwicklungsumgebung.
2. Erstellen Sie in `codecatalyst-eks-source-repository` einen Ordner mit dem Namen `public-html`.
3. Erstellen Sie in `/public-html` eine Datei namens `index.html` mit dem folgenden Inhalt:

```
<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. Geben Sie an der Terminal-Eingabeaufforderung Folgendes ein:

```
cd /projects/codecatalyst-eks-source-repository
```

5. Hinzufügen, Festschreiben und Push:

```
git add .
git commit -m "add public-html/index.html"
git push
```

Das `index.html` wird Ihrem Repository in einem `public-html` Ordner hinzugefügt.

## Dockerfile

Das Dockerfile beschreibt das zu verwendende Basis-Decker-Image und die darauf anzuwendenden Docker-Befehle. [Weitere Informationen zum Dockerfile finden Sie in der Dockerfile-Referenz.](#)


Das hier angegebene Dockerfile gibt an, dass das Apache 2.4-Basisimage (`httpd`) verwendet werden soll. Es enthält auch Anweisungen zum Kopieren einer `index.html` aufgerufenen Quelldatei in einen Ordner auf dem Apache-Server, der Webseiten bereitstellt. Die `EXPOSE` Anweisung in der Dockerfile teilt Docker mit, dass der Container auf Port 80 lauscht.

Um das Dockerfile hinzuzufügen

1. Erstellen Sie in `codecatalyst-eks-source-repository` eine Datei namens `Dockerfile` mit dem folgenden Inhalt:

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

Geben Sie keine Dateierweiterung an.

 **Important**

Das Dockerfile muss sich im Stammordner Ihres Repositorys befinden. Der `Docker build` Befehl des Workflows erwartet, dass es dort vorhanden ist.

## 2. Hinzufügen, Festschreiben und Push:

```
git add .
git commit -m "add Dockerfile"
git push
```

Das Dockerfile wird zu Ihrem Repository hinzugefügt.

### deployment.yaml

In diesem Abschnitt fügen Sie Ihrem Repository eine `deployment.yaml` Datei hinzu. Die `deployment.yaml` Datei ist ein Kubernetes-Manifest, das zwei auszuführende Kubernetes-Ressourcentypen oder -arten definiert: einen „Dienst“ und eine „Bereitstellung“.

- Der „Service“ stellt einen Load Balancer in Amazon EC2 bereit. Der Load Balancer stellt Ihnen eine mit dem Internet verbundene öffentliche URL und einen Standardport (Port 80) zur Verfügung, über den Sie zu „Hello, World!“ navigieren können Anwendung.
- Bei der „Bereitstellung“ werden drei Pods bereitgestellt, und jeder Pod enthält einen Docker-Container mit dem Namen „Hello, World!“ Anwendung. Die drei Pods werden auf den Knoten bereitgestellt, die bei der Erstellung des Clusters erstellt wurden.

Das Manifest in diesem Tutorial ist kurz. Ein Manifest kann jedoch eine beliebige Anzahl von Kubernetes-Ressourcentypen wie Pods, Jobs, Ingresses und Netzwerkrichtlinien enthalten. Außerdem können Sie mehrere Manifestdateien verwenden, wenn Ihre Bereitstellung komplex ist.

Um eine `deployment.yaml`-Datei hinzuzufügen

1. Erstellen Sie in einen `codecatalyst-eks-source-repository` Ordner mit dem Namen `Kubernetes`
2. Erstellen Sie in `/Kubernetes` eine Datei namens `deployment.yaml` mit dem folgenden Inhalt:

```
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. Hinzufügen, Festschreiben und Push:

```
git add .
```

```
git commit -m "add Kubernetes/deployment.yaml"
git push
```

Die `deployment.yaml` Datei wird Ihrem Repository in einem Ordner mit dem Namen `hinzugefügtKubernetes`.

Sie haben jetzt alle Ihre Quelldateien hinzugefügt.

Nehmen Sie sich einen Moment Zeit, um Ihre Arbeit zu überprüfen und sicherzustellen, dass Sie alle Dateien in den richtigen Ordnern abgelegt haben. Die Ordnerstruktur ist wie folgt:

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
  |   |– index.html
  |– Dockerfile
```

## Schritt 5: AWS Rollen erstellen

In diesem Abschnitt erstellen Sie AWS IAM-Rollen, die Ihr CodeCatalyst Workflow benötigt, um zu funktionieren. Diese Rollen sind:

- **Build-Rolle** — Erteilt der CodeCatalyst Build-Aktion (im Workflow) die Berechtigung, auf Ihr AWS Konto zuzugreifen und in Amazon ECR und Amazon EC2 zu schreiben.
- **Rolle bereitstellen** — Erteilt der Cluster-Aktion CodeCatalyst Deploy to Kubernetes (im Workflow) die Berechtigung, auf Ihr AWS Konto und Amazon EKS zuzugreifen.

Weitere Informationen zu IAM-Rollen finden Sie unter [IAM-Rollen im Benutzerhandbuch](#). AWS Identity and Access Management

### Note

Um Zeit zu sparen, können Sie anstelle der beiden zuvor aufgeführten Rollen eine einzelne `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle, die so genannte Rolle, erstellen. Weitere Informationen finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle

über sehr umfangreiche Berechtigungen verfügt, was ein Sicherheitsrisiko darstellen kann. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger wichtig ist. In diesem Tutorial wird davon ausgegangen, dass Sie die beiden zuvor aufgeführten Rollen erstellen.

Führen Sie die folgenden Verfahren aus, um die Build- und Deploy-Rollen zu erstellen.

## 1. Um eine Vertrauensrichtlinie für beide Rollen zu erstellen

1. Gehen Sie zu Ihrer Entwicklungsumgebung.
2. Erstellen Sie im Cloud9-*long-string* Verzeichnis eine Datei namens `codecatalyst-eks-trust-policy.json` mit dem folgenden Inhalt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

## 2. Um die Build-Richtlinie für die Build-Rolle zu erstellen

- Erstellen Sie im Cloud9-*long-string* Verzeichnis eine Datei namens `codecatalyst-eks-build-policy.json` mit dem folgenden Inhalt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
        "Effect": "Allow",
        "Action": [
            "ecr:*",
            "ec2:*"
        ],
        "Resource": "*"
    }
]
```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"

```

### 3. Um die Bereitstellungsrichtlinie für die Bereitstellungsrolle zu erstellen

- Erstellen Sie im Cloud9-*long-string* Verzeichnis eine Datei namens `codecatalyst-eks-deploy-policy.json` mit dem folgenden Inhalt:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}
```

**Note**

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinien-Anweisung und grenzen Sie dann die Richtlinie mit dem Ressourcennamen ab, sobald sie verfügbar ist.

```
"Resource": "*"
```

Sie haben Ihrer Entwicklungsumgebung jetzt drei Richtliniendokumente hinzugefügt. Ihre Verzeichnisstruktur sieht jetzt wie folgt aus:

```
|– 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. Um die Build-Richtlinie hinzuzufügen AWS

1. Geben Sie im Dev Environment-Terminal Folgendes ein:

```
cd /projects
```

2. Geben Sie ein:

```
aws iam create-policy \  
  --policy-name codecatalyst-eks-build-policy \  
  --policy-document file://codecatalyst-eks-build-policy.json
```

3. Drücken Sie die Eingabetaste.
4. Notieren Sie sich den "arn": Wert in der Befehlsausgabe, zum Beispiel `arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy`. Sie benötigen diesen ARN später.

## 5. Um die Bereitstellungsrichtlinie hinzuzufügen AWS

1. Geben Sie ein:

```
aws iam create-policy \  
  --policy-name codecatalyst-eks-deploy-policy \  
  --policy-document file://codecatalyst-eks-deploy-policy.json
```

2. Drücken Sie die Eingabetaste.
3. Notieren Sie sich in der Befehlsausgabe den "arn": Wert der Bereitstellungsrichtlinie, `arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy` z. B. Sie benötigen diesen ARN später.

## 6. Um die Build-Rolle zu erstellen

1. Geben Sie ein:

```
aws iam create-role \  
  --role-name codecatalyst-eks-build-role \  
  --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Drücken Sie die Eingabetaste.
3. Geben Sie ein:

```
aws iam attach-role-policy \  
  --role-name codecatalyst-eks-build-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy
```

Wobei *arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy* durch den ARN der Build-Richtlinie ersetzt wird, die Sie zuvor notiert haben.

4. Drücken Sie die Eingabetaste.
5. Geben Sie an der Terminal-Eingabeaufforderung Folgendes ein:

```
aws iam get-role \  
  --role-name codecatalyst-eks-build-role
```

6. Drücken Sie die Eingabetaste.



7. Notieren Sie sich den "Arn": Wert der Rolle, zum Beispiel `arn:aws:iam::111122223333:role/codecatalyst-eks-build-role`. Sie benötigen diesen ARN später.

## 7. Um die Bereitstellungsrolle zu erstellen

1. Geben Sie ein:

```
aws iam create-role \  
  --role-name codecatalyst-eks-deploy-role \  
  --assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Drücken Sie die Eingabetaste.

3. Geben Sie ein:

```
aws iam attach-role-policy \  
  --role-name codecatalyst-eks-deploy-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy
```

Wobei *arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy* durch den ARN der Bereitstellungsrichtlinie ersetzt wird, die Sie zuvor notiert haben.

4. Drücken Sie die Eingabetaste.

5. Geben Sie ein:

```
aws iam get-role \  
  --role-name codecatalyst-eks-deploy-role
```

6. Drücken Sie die Eingabetaste.

7. Notieren Sie sich den "Arn": Wert der Rolle, zum Beispiel `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role`. Sie benötigen diesen ARN später.

Sie haben jetzt Build- und Deploy-Rollen erstellt und deren ARNs notiert.

## Schritt 6: AWS Rollen hinzufügen CodeCatalyst

In diesem Schritt fügen Sie die Build-Rolle (`codecatalyst-eks-build-role`) und die Bereitstellungsrolle (`codecatalyst-eks-deploy-role`) zu der Rolle hinzu AWS-Konto , die Sie

mit Ihrem Bereich verbunden haben. Dadurch sind die Rollen für die Verwendung in Ihrem Workflow verfügbar.

Um Build- und Deploy-Rollen zu Ihrem hinzuzufügen AWS-Konto

1. Navigieren Sie in der CodeCatalyst Konsole zu Ihrem Bereich.
2. Wählen Sie oben Einstellungen aus.
3. Wählen Sie im Navigationsbereich AWS Konten aus. Eine Liste von Konten wird angezeigt.
4. Kopieren Sie in die Spalte CodeCatalyst Amazon-Anzeigenname den Anzeigenamen der AWS-Konto Stelle, in der Sie Ihre Build- und Deploy-Rollen erstellt haben. (Es könnte eine Zahl sein.) Sie benötigen diesen Wert später, wenn Sie Ihren Workflow erstellen.
5. Wählen Sie den Anzeigenamen.
6. Wählen Sie in der AWS Managementkonsole die Option Rollen verwalten aus.

Die Seite „IAM-Rolle zu Amazon CodeCatalyst Space hinzufügen“ wird angezeigt. Möglicherweise müssen Sie sich anmelden, um auf die Seite zuzugreifen.

7. Wählen Sie Eine bestehende Rolle hinzufügen, die Sie in IAM erstellt haben.

Eine Dropdownliste wird angezeigt. In der Liste werden die Build- und Deploy-Rollen sowie alle anderen IAM-Rollen mit einer Vertrauensrichtlinie angezeigt, die auch die Dienstprinzipale `codecatalyst-runner.amazonaws.com` und die `codecatalyst.amazonaws.com` Dienstprinzipale umfasst.

8. Fügen Sie aus der Dropdownliste Folgendes hinzu:
  - `codecatalyst-eks-build-role`
  - `codecatalyst-eks-deploy-role`

#### Note

Wenn Sie das sehen `The security token included in the request is invalid`, liegt es möglicherweise daran, dass Sie nicht über die richtigen Berechtigungen verfügen. Um dieses Problem zu beheben, melden Sie sich ab und melden Sie sich mit dem AWS Konto wieder an, das Sie bei der Erstellung Ihres CodeCatalyst Bereichs verwendet haben. AWS

9. Kehren Sie zur CodeCatalyst Konsole zurück und aktualisieren Sie die Seite.

Die Rollen Build und Deploy sollten jetzt unter IAM-Rollen angezeigt werden.

Diese Rollen sind jetzt für die Verwendung in CodeCatalyst Workflows verfügbar.

## Schritt 7: Aktualisieren Sie das ConfigMap

Sie müssen die Bereitstellungsrolle, die Sie in [Schritt 5: AWS Rollen erstellen](#) der ConfigMap Kubernetes-Datei erstellt haben, hinzufügen, damit die Cluster-Aktion Deploy to Kubernetes (in Ihrem Workflow) auf Ihren Cluster zugreifen und mit ihm interagieren kann. Sie können `eksctl` oder verwenden, `kubect1` um diese Aufgabe auszuführen.

Um die ConfigMap Kubernetes-Datei mit `eksctl` zu konfigurieren

- Geben Sie im Dev Environment-Terminal Folgendes ein:

```
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
```

Wobei gilt:

- *codecatalyst-eks-cluster* wird durch den Clusternamen des Amazon EKS-Clusters ersetzt.
- *arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role* wird durch den ARN der Bereitstellungsrolle ersetzt, in der Sie erstellt haben. [Schritt 5: AWS Rollen erstellen](#)
- *codecatalyst-eks-deploy-role* (neben) wird durch den Namen der Bereitstellungsrolle ersetzt, in der Sie sie `--username` erstellt haben. [Schritt 5: AWS Rollen erstellen](#)

### Note

Wenn Sie sich entschieden haben, keine Bereitstellungsrolle zu erstellen, *codecatalyst-eks-deploy-role* ersetzen Sie sie durch den Namen der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle. Weitere Informationen über diese Rolle finden Sie unter [Schritt 5: AWS Rollen erstellen](#).

- *us-west-2* wird durch Ihre Region ersetzt.

Einzelheiten zu diesem Befehl finden Sie unter [IAM-Benutzer und -Rollen verwalten](#).

Eine Meldung ähnlich der folgenden wird angezeigt:

```
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
```

Um die ConfigMap Kubernetes-Datei mit kubectl zu konfigurieren

1. Geben Sie im Dev Environment-Terminal Folgendes ein:

```
kubectl edit configmap -n kube-system aws-auth
```

Die ConfigMap Datei wird auf dem Bildschirm angezeigt.

2. Fügen Sie den Text in roter Kursivschrift hinzu:

```
# 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:
  ...
```

Wobei gilt:

- `arn:aws:iam: :111122223333:role/ codecatalyst-eks-deploy-role` wird durch den ARN der Bereitstellungsrolle ersetzt, in der Sie erstellt haben. [Schritt 5: AWS Rollen erstellen](#)
- `codecatalyst-eks-deploy-role`(neben) wird durch den Namen der Bereitstellungsrolle ersetzt, in der Sie sie `username:` erstellt haben. [Schritt 5: AWS Rollen erstellen](#)

#### Note

Wenn Sie sich entschieden haben, keine Bereitstellungsrolle zu erstellen, `codecatalyst-eks-deploy-role` ersetzen Sie sie durch den Namen der CodeCatalystWorkflowDevelopmentRole-`spaceName` Rolle. Weitere Informationen über diese Rolle finden Sie unter [Schritt 5: AWS Rollen erstellen](#).

Einzelheiten finden Sie unter [Aktivieren des IAM-Prinzipalzugriffs auf Ihren Cluster](#) im Amazon EKS-Benutzerhandbuch.

Sie haben jetzt der Bereitstellungsrolle und damit der Aktion Deploy to Amazon EKS `system:masters` Berechtigungen für Ihren Kubernetes-Cluster erteilt.

## Schritt 8: Erstellen Sie einen Workflow und führen Sie ihn aus

In diesem Schritt erstellen Sie einen Workflow, der Ihre Quelldateien zu einem Docker-Image zusammenbaut und das Image dann in Tree-Pods in Ihrem Amazon EKS-Cluster bereitstellt.

Der Workflow besteht aus den folgenden Bausteinen, die nacheinander ausgeführt werden:

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).


- Eine Build-Aktion (BuildBackend) — Beim Auslösen erstellt die Aktion das Docker-Image mithilfe der Dockerfile und überträgt das Image an Amazon ECR. Die Build-Aktion aktualisiert auch die `$IMAGE_TAG` Variablen `$REPOSITORY_URI` und in der `deployment.yaml` Datei mit den richtigen Werten und erstellt dann ein Ausgabeartefakt aus dieser Datei und allen anderen Dateien im Ordner. Kubernetes In diesem Tutorial ist die einzige Datei im Kubernetes Ordner, `deployment.yaml` aber Sie könnten weitere Dateien hinzufügen. Das Artefakt wird als Eingabe für die Bereitstellungsaktion verwendet, die als Nächstes folgt.

Weitere Informationen zur Build-Aktion finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#).

- Eine Bereitstellungsaktion (DeployToEKS) — Nach Abschluss der Build-Aktion sucht die Bereitstellungsaktion nach dem von der Build-Aktion (Manifests) generierten Ausgabeartefakt und findet die darin enthaltene `deployment.yaml` Datei. Die Aktion folgt dann den Anweisungen in der `deployment.yaml` Datei, um drei Pods auszuführen, von denen jeder ein einzelnes „Hello, World!“ enthält Docker-Container — in Ihrem Amazon EKS-Cluster.

So erstellen Sie ein Workflow

1. Gehen Sie zur Konsole. CodeCatalyst
2. Navigiere zu deinem Projekt (`codecatalyst-eks-project`).
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie Workflow erstellen aus.
5. Wählen Sie für Quell-Repository die Option `codecatalyst-eks-source-repository`.
6. Wählen Sie für Branch die Option `main`.
7. Wählen Sie Erstellen.
8. Löschen Sie den YAML-Beispielcode.
9. Fügen Sie den folgenden YAML-Code hinzu, um eine neue Workflow-Definitionsdatei zu erstellen:

 Note

Weitere Informationen zur Workflow-Definitionsdatei finden Sie unter [Referenz zur Workflow-Definition](#).

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/


```

Ersetzen Sie im vorherigen Code:

- Beide Instanzen von *codecatalyst-eks-environment* mit dem Namen der Umgebung, in der Sie erstellt haben [Voraussetzungen](#).
- Beide Instanzen von *codecatalyst-account-connection* mit dem Anzeigenamen Ihrer Kontoverbindung. Der Anzeigename kann eine Zahl sein. Weitere Informationen finden Sie unter [Schritt 6: AWS Rollen hinzufügen CodeCatalyst](#).
- *codecatalyst-eks-build-role* mit dem Namen der Build-Rolle, in der Sie sie erstellt haben [Schritt 5: AWS Rollen erstellen](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo* (in der Value: Eigenschaft) mit der URI des Amazon ECR-Repositorys, in dem Sie es erstellt haben. [Schritt 3: Erstellen Sie ein Amazon ECR-Image-Repository](#)



- `111122223333.dkr.ecr.us-west-2.amazonaws.com` (im Run: `aws ecr` Befehl) mit der URI des Amazon ECR-Repositorys ohne das Bildsuffix (`.`). `/codecatalyst-eks-image-repo`
- `codecatalyst-eks-deploy-role` mit dem Namen der Bereitstellungsrolle, in der Sie sie erstellt haben. [Schritt 5: AWS Rollen erstellen](#)
- Beide Instanzen von `us-west-2` mit Ihrem AWS Regionalcode. Eine Liste der Regionalcodes finden Sie unter [Regionale Endpunkte](#) in der. Allgemeine AWS-Referenz

 Note

Wenn Sie sich entschieden haben, keine Build- und Deploy-Rollen zu erstellen, ersetzen Sie `codecatalyst-eks-build-role` und `codecatalyst-eks-deploy-role` durch den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle. Weitere Informationen über diese Rolle finden Sie unter [Schritt 5: AWS Rollen erstellen](#).

10. (Optional) Wählen Sie „Validieren“, um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie ihn festschreiben.
11. Wählen Sie Commit (Übergeben).
12. Geben Sie im Dialogfeld „Workflow bestätigen“ Folgendes ein:
  - a. Entfernen Sie bei Nachricht bestätigen den Text und geben Sie Folgendes ein:

**Add first workflow**

- b. Wählen Sie für Repository `codecatalyst-eks-source-repository`.
- c. Wählen Sie als Branch-Name die Option `main` aus.
- d. Wählen Sie Commit (Übergeben).

Sie haben jetzt einen Workflow erstellt. Eine Workflow-Ausführung wird aufgrund des oben im Workflow definierten Triggers automatisch gestartet. Insbesondere wenn Sie die `workflow.yaml` Datei in Ihr Quell-Repository übertragen (und per Push) haben, hat der Trigger die Workflow-Ausführung gestartet.

## Um den Fortschritt der Workflow-Ausführung zu sehen

1. Wählen Sie im Navigationsbereich der CodeCatalyst Konsole CI/CD und dann Workflows aus.
2. Wählen Sie den Workflow aus, den Sie gerade erstellt haben, `codecatalyst-eks-workflow`.
3. Wählen Sie BuildBackend, ob Sie den Baufortschritt sehen möchten.
4. Wählen Sie DeployToEKS, um den Fortschritt der Bereitstellung zu sehen.

Weitere Informationen zum Anzeigen von Ausführungsdetails finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).

## Um die Bereitstellung zu überprüfen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie links unten Load Balancers aus.
3. Wählen Sie den Load Balancer aus, der als Teil Ihrer Kubernetes-Bereitstellung erstellt wurde. Wenn Sie sich nicht sicher sind, welchen Load Balancer Sie wählen sollen, suchen Sie auf der Registerkarte „Tags“ nach den folgenden Tags:

- `kubernetes.io/service-name`
- `kubernetes.io/cluster/ekstutorialcluster`

4. Wählen Sie den richtigen Load Balancer aus und wählen Sie den Tab Beschreibung aus.
5. Kopieren Sie den DNS-Namenswert und fügen Sie ihn in die Adressleiste Ihres Browsers ein.

Das „Hallo, Welt!“ In Ihrem Browser wird eine Webseite angezeigt, die darauf hinweist, dass Sie Ihre Anwendung erfolgreich bereitgestellt haben.

## Schritt 9: Nehmen Sie eine Änderung an Ihren Quelldateien vor

In diesem Abschnitt nehmen Sie eine Änderung an der `index.html` Datei in Ihrem Quell-Repository vor. Diese Änderung veranlasst den Workflow, ein neues Docker-Image zu erstellen, es mit einer Commit-ID zu kennzeichnen, es an Amazon ECR weiterzuleiten und es in Amazon ECS bereitzustellen.

### Um die Datei `index.html` zu ändern

1. Gehen Sie zu Ihrer Entwicklungsumgebung.
2. Wechseln Sie an der Terminal-Eingabeaufforderung zu Ihrem Quell-Repository:

```
cd /projects/codecatalyst-eks-source-repository
```

3. Rufen Sie die neuesten Workflow-Änderungen ab:

```
git pull
```

4. Öffnen Sie `codecatalyst-eks-source-repository/public-html/index.html`.
5. Ändern Sie in Zeile 14 den `Hello, World! Text in Tutorial complete!`.
6. Hinzufügen, Festschreiben und Push:

```
git add .  
git commit -m "update index.html title"  
git push
```

Eine Workflow-Ausführung wird automatisch gestartet.

7. (Optional) Geben Sie ein:

```
git show HEAD
```

Notieren Sie sich die Commit-ID für die `index.html` Änderung. Diese Commit-ID wird dem Docker-Image zugeordnet, das durch die Workflow-Ausführung bereitgestellt wird, die Sie gerade gestartet haben.

8. Beobachten Sie den Fortschritt der Bereitstellung:
  - a. Wählen Sie in der CodeCatalyst Konsole im Navigationsbereich CI/CD und dann Workflows aus.
  - b. Wählen Sie `codecatalyst-eks-workflow`, ob Sie die letzte Ausführung anzeigen möchten.
  - c. Wählen Sie `DeployToEKS BuildBackend`, um den Fortschritt der Workflow-Ausführung zu sehen.
9. Stellen Sie wie folgt sicher, dass Ihre Anwendung aktualisiert wurde:
  - a. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
  - b. Wählen Sie links unten Load Balancers aus.
  - c. Wählen Sie den Load Balancer aus, der als Teil Ihrer Kubernetes-Bereitstellung erstellt wurde.

- d. Kopieren Sie den DNS-Namenswert und fügen Sie ihn in die Adressleiste Ihres Browsers ein.

Das 'Tutorial ist abgeschlossen!' In Ihrem Browser wird eine Webseite angezeigt, die darauf hinweist, dass Sie erfolgreich eine neue Version Ihrer Anwendung bereitgestellt haben.

10. (Optional) Wechseln Sie in AWS zur Amazon ECR-Konsole und überprüfen Sie, ob das neue Docker-Image mit der Commit-ID aus Schritt 7 dieses Verfahrens gekennzeichnet wurde.

## Bereinigen

Sie sollten Ihre Umgebung bereinigen, damit Ihnen die in diesem Tutorial verwendeten Speicher- und Rechenressourcen nicht unnötig in Rechnung gestellt werden.

So räumen Sie auf

1. Löschen Sie Ihren Cluster:

- Geben Sie im Dev Environment-Terminal Folgendes ein:

```
eksctl delete cluster --region=us-west-2 --name=codecatalyst-eks-cluster
```

Wobei gilt:

- *us-west-2* wird durch Ihre Region ersetzt.
- *codecatalyst-eks-cluster* wird durch den Namen des Clusters ersetzt, den Sie erstellt haben.

Nach 5-10 Minuten werden der Cluster und die zugehörigen Ressourcen gelöscht, einschließlich, aber nicht beschränkt auf AWS CloudFormation Stacks, Knotengruppen (in Amazon EC2) und Load Balancer.

### Important

Wenn der `eksctl delete cluster` Befehl nicht funktioniert, müssen Sie möglicherweise Ihre AWS Anmeldeinformationen oder Ihre Anmeldeinformationen aktualisieren. `kubectl` Wenn Sie sich nicht sicher sind, welche Anmeldeinformationen Sie aktualisieren sollen, aktualisieren Sie zuerst die AWS Anmeldeinformationen.

Informationen zum Aktualisieren Ihrer AWS Anmeldeinformationen finden Sie unter [Wie behebe ich die Fehler „Anmeldeinformationen konnten nicht gefunden werden“ und ExpiredToken „?“](#). Informationen zum Aktualisieren Ihrer kubectl Anmeldeinformationen finden Sie unter [Wie behebe ich die Fehler „Es konnte keine Verbindung zum Server hergestellt werden“?](#).

2. Bereinigen Sie in der AWS Konsole wie folgt:
  1. Löschen `codecatalyst-eks-image-repo` Sie in Amazon ECR.
  2. Löschen Sie im IAM Identity Center:
    - a. `codecatalyst-eks-user`
    - b. `codecatalyst-eks-permission-set`
  3. Löschen Sie in IAM:
    - `codecatalyst-eks-build-role`
    - `codecatalyst-eks-deploy-role`
    - `codecatalyst-eks-build-policy`
    - `codecatalyst-eks-deploy-policy`
3. Bereinigen Sie in der CodeCatalyst Konsole wie folgt:
  1. Löschen `codecatalyst-eks-workflow`.
  2. Löschen `codecatalyst-eks-environment`.
  3. Löschen `codecatalyst-eks-source-repository`.
  4. Löschen Sie Ihre Entwicklungsumgebung.
  5. Löschen `codecatalyst-eks-project`.

In diesem Tutorial haben Sie gelernt, wie Sie mithilfe eines CodeCatalyst Workflows und einer Cluster-Aktion „Deploy to Kubernetes“ eine Anwendung für einen Amazon EKS-Service bereitstellen.

## Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen

### Tip

Ein Tutorial, das Ihnen zeigt, wie Sie die Aktion `AWS CloudFormation Stack bereitstellen` verwenden, finden Sie unter [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#).

In diesem Abschnitt wird beschrieben, wie Sie die Aktion `AWS CloudFormation Stack bereitstellen` zu Ihrem Workflow hinzufügen. Die Aktion erstellt einen CloudFormation Stack in, der auf einer von Ihnen bereitgestellten Vorlage AWS basiert. Die Vorlage kann eine sein:

- `AWS CloudFormation Vorlage` — Weitere Informationen finden Sie unter [Mit AWS CloudFormation Vorlagen arbeiten](#).
- `AWS SAM Vorlage` — Weitere Informationen finden Sie in der [AWS Serverless Application Model \(AWS SAM\) -Spezifikation](#).

### Note

Um eine `AWS SAM Vorlage` verwenden zu können, müssen Sie zuerst Ihre `AWS SAM Anwendung` mithilfe des `sam package` Vorgangs paketieren. Ein Tutorial, das Ihnen zeigt, wie Sie dieses Verpacken im Rahmen eines CodeCatalyst Amazon-Workflows automatisch durchführen, finden Sie unter [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#).

Wenn der Stapel bereits vorhanden ist, führt die Aktion den CloudFormation `CreateChangeSet` Vorgang und dann den `ExecuteChangeSet` Vorgang aus. Die Aktion wartet dann auf die Bereitstellung der Änderungen und markiert sich je nach Ergebnis entweder als erfolgreich oder als fehlgeschlagen.

Verwenden Sie die Aktion `AWS CloudFormation Stack bereitstellen`, wenn Sie bereits über eine `AWS CloudFormation AWS SAM Oder-Vorlage` verfügen, die Ressourcen enthält, die Sie bereitstellen möchten, oder wenn Sie planen, im Rahmen einer [Workflow-Build-Aktion](#) mithilfe von Tools wie `AWS SAM` und [AWS Cloud Development Kit \(AWS CDK\)](#) automatisch eine Vorlage zu generieren.

Es gibt keine Einschränkungen hinsichtlich der Vorlage, die Sie verwenden können — unabhängig davon, welche Vorlage Sie verwenden können, CloudFormation oder die AWS SAM. Sie können sie zusammen mit der Aktion „AWS CloudFormation Stack bereitstellen“ verwenden können.

## Visual

Um die Aktion „AWS CloudFormation Stapel bereitstellen“ mit dem visuellen Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Bearbeiten aus.
  6. Wählen Sie Visual.
  7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
  9. Suchen Sie nach der Aktion AWS CloudFormation Stack bereitstellen und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Oder
- Wählen Sie AWS CloudFormation Stack bereitstellen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „Stack bereitstellen AWS CloudFormation“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.

11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um die Aktion „AWS CloudFormation Stack bereitstellen“ mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Bearbeiten aus.
  6. Wählen Sie YAML.
  7. Wählen Sie links oben + Aktionen, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
  9. Suchen Sie nach der Aktion AWS CloudFormation Stack bereitstellen und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Oder
- Wählen Sie AWS CloudFormation Stack bereitstellen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
  10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“](#).
  11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.



12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

Variablen, die durch die Aktion „AWS CloudFormation Stack bereitstellen“ erzeugt wurden

Wenn die Aktion AWS CloudFormation Stack bereitstellen ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [Aktionsvariablen „AWS CloudFormationStack bereitstellen“](#) in der [Liste der vordefinierten Variablen](#).

Definition der Aktion „AWS CloudFormation Stack bereitstellen“

Die Aktion „AWS CloudFormation Stack bereitstellen“ ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“](#) in der [Referenz zur Workflow-Definition](#).

Aktion „In Amazon ECS bereitstellen“ hinzufügen

 Tip

Ein Tutorial, das Ihnen zeigt, wie Sie die Aktion Deploy to Amazon ECS verwenden, finden Sie unter [Tutorial: Bereitstellen einer Anwendung in Amazon ECS](#).

 Tip

Um ein funktionierendes Beispiel für die Aktion Deploy to Amazon ECS zu erhalten, erstellen Sie ein Projekt entweder mit der Node.js API mit AWS Fargate oder der Java-API mit AWS Fargate Blueprint. Weitere Informationen finden Sie unter [Ein Projekt mit einem Blueprint erstellen](#).

In diesem Abschnitt wird beschrieben, wie Sie die Aktion Deploy to Amazon ECS zu Ihrem Workflow hinzufügen. Diese Aktion registriert eine von Ihnen bereitgestellte [Aufgabendefinitionsdatei](#). Nach der Registrierung wird die Aufgabendefinition dann von Ihrem [Amazon ECS-Service instanziiert, der in Ihrem Amazon ECS-Cluster](#) ausgeführt wird. Das „Instanziiieren einer Aufgabendefinition“ entspricht der Bereitstellung einer Anwendung in Amazon ECS.

Um diese Aktion verwenden zu können, müssen Sie über einen vorhandenen Amazon ECS-Cluster, einen Service und eine Aufgabendefinitionsdatei verfügen.

Weitere Informationen zu Amazon ECS finden Sie im Amazon Elastic Container Service Developer Guide.

## Visual

Um die Aktion „In Amazon ECS bereitstellen“ mit dem visuellen Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der Aktion Deploy to Amazon ECS und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie Deploy to Amazon ECS. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „Deploy to Amazon ECS“](#). Diese Referenz enthält detaillierte Informationen zu jedem

Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.

11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um die Aktion „In Amazon ECS bereitstellen“ mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Bearbeiten aus.
  6. Wählen Sie YAML.
  7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
  9. Suchen Sie nach der Aktion Deploy to Amazon ECS und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Oder
- Wählen Sie Deploy to Amazon ECS. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „Deploy to Amazon ECS“](#).

11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Variablen, die durch die Aktion „In Amazon ECS bereitstellen“ erzeugt wurden

Wenn die Aktion Deploy to Amazon ECS ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [Aktionsvariablen „In Amazon ECS bereitstellen“](#) in der [Liste der vordefinierten Variablen](#).

## Aktionsdefinition „In Amazon ECS bereitstellen“

Die Aktion In Amazon ECS bereitstellen ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „Deploy to Amazon ECS“](#) in der [Referenz zur Workflow-Definition](#).

## Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen

### Tip

Ein Tutorial, das Ihnen zeigt, wie Sie die Cluster-Aktion Deploy to Kubernetes verwenden, finden Sie unter [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#)

In diesem Abschnitt wird beschrieben, wie Sie die Cluster-Aktion „In Kubernetes bereitstellen“ zu Ihrem Workflow hinzufügen. Diese Aktion stellt Ihre Anwendung in einem Kubernetes-Cluster bereit, den Sie in Amazon Elastic Kubernetes Service (EKS) eingerichtet haben. Dabei werden eine oder mehrere Kubernetes-Manifestdateien verwendet, die Sie bereitstellen. Ein Beispielmanifest finden Sie unter [deployment.yaml](#) [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#)

Weitere Informationen zu Kubernetes finden Sie in der [Kubernetes-Dokumentation](#).

Weitere Informationen zu Amazon EKS finden Sie unter [Was ist Amazon EKS?](#) im Amazon EKS-Benutzerhandbuch.

## Funktionsweise

Der Cluster „Auf Kubernetes bereitstellen“ funktioniert wie folgt:

1. Zur Laufzeit installiert die Aktion das `kubectl` Kubernetes-Hilfsprogramm auf dem CodeCatalyst Computer, auf dem die Aktion ausgeführt wird. Die Aktion ist so konfiguriert `kubectl`, dass sie auf den Amazon EKS-Cluster verweist, den Sie bei der Konfiguration der Aktion angegeben haben. Das `kubectl` Hilfsprogramm ist erforderlich, um den `kubectl apply` Befehl als Nächstes auszuführen.
2. Die Aktion führt den `kubectl apply -f my-manifest.yaml` Befehl aus, der die Anweisungen in `my-manifest.yaml` ausführt, um Ihre Anwendung als Satz von Containern und Pods im konfigurierten Cluster bereitzustellen. Weitere Informationen zu diesem Befehl finden Sie im Thema [kubectl apply in der Kubernetes-Referenzdokumentation](#).

## Voraussetzungen

Um diese Aktion verwenden zu können, müssen Sie Folgendes vorbereitet haben:

### Tip

Folgen Sie den Anweisungen unter, um diese Voraussetzungen schnell einzurichten [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#).

- Ein Kubernetes-Cluster in Amazon EKS. Informationen zu Clustern finden Sie unter [Amazon EKS-Cluster](#) im Amazon EKS-Benutzerhandbuch.
- Mindestens ein Dockerfile, das beschreibt, wie Sie Ihre Anwendung zu einem Docker-Image zusammenstellen. [Weitere Informationen zu Dockerfiles finden Sie in der Dockerfile-Referenz](#).
- Mindestens eine Kubernetes-Manifestdatei, die in der Kubernetes-Dokumentation als Konfigurationsdatei oder Konfiguration bezeichnet wird. Weitere Informationen finden Sie in der [Kubernetes-Dokumentation unter Ressourcen verwalten](#).
- Eine IAM-Rolle, die der Cluster-Aktion Deploy to Kubernetes die Möglichkeit gibt, auf Ihren Amazon EKS-Cluster zuzugreifen und mit ihm zu interagieren. Weitere Informationen erhalten Sie unter dem Thema [Role](#) im [Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#).

Nachdem Sie diese Rolle erstellt haben, müssen Sie sie hinzufügen zu:

- Ihre Kubernetes-Datei ConfigMap . Informationen zum Hinzufügen einer Rolle zu einer ConfigMap Datei finden Sie unter [Aktivieren des IAM-Prinzipalzugriffs auf Ihren Cluster](#) im Amazon EKS-Benutzerhandbuch.

- CodeCatalyst. Informationen zum Hinzufügen einer IAM-Rolle zu finden Sie CodeCatalyst unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).
- Ein CodeCatalyst Raum, ein Projekt und eine Umgebung. Der Bereich und die Umgebung müssen beide mit dem AWS Konto verbunden sein, in dem Sie Ihre Anwendung bereitstellen werden. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#), [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#) und [Arbeiten mit Umgebungen](#).
- Ein Quell-Repository, das von unterstützt wird CodeCatalyst. Das Repository speichert Ihre Anwendungsquelldateien, Dockerfiles und Kubernetes-Manifeste. Weitere Informationen finden Sie unter [Quell-Repositoryn in CodeCatalyst](#).

## Die Aktion „Auf Cluster bereitstellen“ hinzufügen

Verwenden Sie die folgenden Anweisungen, um die Aktion In Cluster bereitstellen zu Ihrem Workflow hinzuzufügen.

### Visual

Um die Aktion „Auf Kubernetes-Cluster bereitstellen“ mit dem visuellen Editor hinzuzufügen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der Cluster-Aktion Deploy to Kubernetes und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie Auf Kubernetes-Cluster bereitstellen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld gehen Sie wie folgt vor:
  - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- 10. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
- 11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
- 12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um die Aktion „Auf Kubernetes-Cluster bereitstellen“ mit dem YAML-Editor hinzuzufügen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der Cluster-Aktion Deploy to Kubernetes und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie **Auf Kubernetes-Cluster bereitstellen** aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld gehen Sie wie folgt vor:
  - (Optional) Wählen Sie **Herunterladen**, um [den Quellcode der Aktion anzuzeigen](#).
  - Wählen Sie **Zum Workflow hinzufügen**, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- 10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#).
- 11. (Optional) Wählen Sie **Validieren**, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
- 12. Wählen Sie **Commit**, geben Sie eine Commit-Nachricht ein und wählen Sie erneut **Commit**.

Variablen, die durch die Aktion **„Auf Kubernetes-Cluster bereitstellen“** erzeugt wurden

Wenn die Cluster-Aktion **„Auf Kubernetes bereitstellen“** ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [Aktionsvariablen „Auf Kubernetes-Cluster bereitstellen“](#) in der [Liste der vordefinierten Variablen](#)

## Aktionsdefinition **„Auf Kubernetes-Cluster bereitstellen“**

Die Cluster-Aktion **„Auf Kubernetes bereitstellen“** ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#) in der [Referenz zur Workflow-Definition](#)

## Aktion **„AWS CDK Deploy“** hinzufügen

In diesem Abschnitt wird beschrieben, wie Sie die Aktion **„AWS CDK Bereitstellen“** zu Ihrem Workflow hinzufügen. Die AWS CDK Bereitstellungsaktion synthetisiert Ihre AWS Cloud Development Kit (AWS CDK) App und stellt sie bereit. Wenn Ihre App bereits in vorhanden ist AWS, wird sie bei Bedarf durch die Aktion aktualisiert.

Allgemeine Informationen zum Schreiben von Apps mit dem AWS CDK finden Sie unter [Was ist der AWS CDK?](#) im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch.

### Themen

- [Wann sollte diese Aktion verwendet werden](#)



- [Funktionsweise](#)
- [CDK-CLI-Versionen, die von der Aktion „AWS CDK Deploy“ verwendet werden](#)
- [Wie viele Stapel kann die Aktion bereitstellen?](#)
- [Voraussetzungen](#)
- [Beispiel-ArbeitsFlows](#)
- [Die Aktion „Bereitstellen“AWS CDK wird hinzugefügt](#)
- [Variablen, die durch die Aktion „AWS CDK Deploy“ erzeugt wurden](#)
- [Definition der Aktion „AWS CDK bereitstellen“](#)

## Wann sollte diese Aktion verwendet werden

Verwenden Sie diese Aktion AWS CDK, wenn Sie mit dem eine App entwickelt haben und diese nun automatisch als Teil des automatisierten Workflows für kontinuierliche Integration und Bereitstellung (CI/CD) bereitstellen möchten. Möglicherweise möchten Sie Ihre AWS CDK App beispielsweise automatisch bereitstellen, wenn jemand eine Pull-Anfrage zusammenführt, die sich auf Ihre AWS CDK App-Quelle bezieht.

## Funktionsweise

Die AWS CDK Bereitstellung funktioniert wie folgt:

1. [Wenn Sie zur Laufzeit Version 1.0.12 oder früher der Aktion angegeben haben, lädt die Aktion die neueste CDK-CLI \(auch AWS CDK Toolkit genannt\) in das Build-Image herunter. CodeCatalyst](#)

Wenn Sie Version 1.0.13 oder höher angegeben haben, ist die Aktion mit einer [bestimmten Version](#) der CDK-CLI gebündelt, sodass kein Download stattfindet.

2. Die Aktion verwendet die CDK-CLI, um den `cdk deploy` Befehl auszuführen. Dieser Befehl synthetisiert Ihre AWS CDK App und stellt sie bereit in. AWSWeitere Informationen zu diesem Befehl finden Sie im Developer Guide unter dem Thema [AWS CDK Toolkit \(cdk-Befehl\)](#).AWS Cloud Development Kit (AWS CDK)

## CDK-CLI-Versionen, die von der Aktion „AWS CDK Deploy“ verwendet werden

Die folgende Tabelle zeigt, welche Version der CDK-CLI standardmäßig von verschiedenen Versionen der AWS CDK Bereitstellungsaktion verwendet wird.

**Note**

Möglicherweise können Sie die Standardeinstellung überschreiben. Weitere Informationen finden Sie unter [CdkCliVersion](#) im [Referenz zur Aktion „AWS CDK Bereitstellen“](#).

Aktionsversion „AWS CDK bereitstellen“	AWS CDK CLI-Version
1.0.0 — 1.0.12	brandneue
1.0.13 oder später	2.99.1

## Wie viele Stapel kann die Aktion bereitstellen?

Bei der AWS CDK Bereitstellung kann nur ein einziger Stack bereitgestellt werden. Wenn Ihre AWS CDK App aus mehreren Stacks besteht, müssen Sie einen übergeordneten Stack mit verschachtelten Stacks erstellen und den übergeordneten Stapel mithilfe dieser Aktion bereitstellen.

## Voraussetzungen

Bevor Sie die Aktion „AWS CDK Bereitstellen“ verwenden können, müssen Sie die folgenden Aufgaben ausführen:

- Halten Sie eine AWS CDK App bereit. Sie können Ihre AWS CDK App mit AWS CDK Version 1 oder 2 in jeder Programmiersprache schreiben, die von der unterstützt wird AWS CDK. Stellen Sie sicher, dass Ihre AWS CDK App-Dateien verfügbar sind in:
  - Ein CodeCatalyst [Quell-Repository](#) oder
  - Ein CodeCatalyst [Ausgabeartefakt](#), das durch eine andere Workflow-Aktion generiert wurde
- Bootstrap für Ihre Umgebung AWS . Um ein Bootstrap zu erstellen, können Sie:
  - Verwenden Sie eine der Methoden, die im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch unter [How to Bootstrap](#) beschrieben sind.
  - Verwenden Sie die AWS CDK Bootstrap-Aktion. Sie können diese Aktion im selben Workflow wie Ihre AWS CDK Bereitstellung oder in einem anderen Workflow hinzufügen. Stellen Sie einfach sicher, dass die Bootstrap-Aktion mindestens einmal ausgeführt wird, bevor AWS CDK Sie die Bereitstellungsaktion ausführen, damit die erforderlichen Ressourcen vorhanden sind.

Weitere Informationen zur AWS CDK Bootstrap-Aktion finden Sie unter [Aktion „AWS CDK Bootstrap“ hinzufügen](#)

Weitere Informationen zu Bootstrapping finden Sie unter [Bootstrapping](#) im Entwicklerhandbuch.AWS Cloud Development Kit (AWS CDK)

## Beispiel-ArbeitsFlows

Der folgende Beispiel-Workflow umfasst die AWS CDK Bereitstellungsaktion zusammen mit der AWS CDK Bootstrap-Aktion. Der Workflow besteht aus den folgenden Bausteinen, die nacheinander ausgeführt werden:

### Note

Das folgende Workflow-Beispiel dient der Veranschaulichung und funktioniert ohne zusätzliche Konfiguration nicht. Es soll Ihnen ein Beispiel dafür geben, wie ein Workflow aussehen könnte, wenn er mit der Aktion „AWS CDK Bereitstellen“ konfiguriert wird.

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Dieses Repository enthält Ihre AWS CDK App. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine AWS CDK Bootstrap-Aktion (CDKBootstrap) — Beim Auslösen stellt die Aktion den CDKToolkit Bootstrap-Stack in bereit. AWS Wenn der CDKToolkit Stack bereits in der Umgebung vorhanden ist, wird er bei Bedarf aktualisiert. Andernfalls passiert nichts und die Aktion wird als erfolgreich markiert.
- Eine AWS CDK Bereitstellungsaktion (AWS CDK Deploy) — Nach Abschluss der AWS CDK Bootstrap-Aktion synthetisiert die AWS CDK Bereitstellungsaktion Ihren AWS CDK App-Code in einer AWS CloudFormation Vorlage und stellt den in der Vorlage definierten Stack bereit. AWS

```
Name: codecatalyst-cdk-deploy-workflow
```

```
SchemaVersion: 1.0
```

```
Triggers:
```

```
- Type: PUSH
```

```
Branches:
```

```
- main
```

```
Actions:
  CDKBootstrap:
    Identifier: aws/cdk-bootstrap@v1
    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@v1
    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
```

## Die Aktion „Bereitstellen“ AWS CDK wird hinzugefügt

Verwenden Sie die folgenden Anweisungen, um die Aktion „AWS CDK Bereitstellen“ zu Ihrem Workflow hinzuzufügen.

### Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die unter beschriebenen Aufgaben abgeschlossen haben [Voraussetzungen](#).

## Visual

Um die Aktion „AWS CDK Bereitstellen“ mit dem visuellen Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der AWS CDK Bereitstellungsaktion und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie AWS CDK Bereitstellen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld gehen Sie wie folgt vor:
    - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „AWS CDK Bereitstellen“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
  11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
  12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

**Note**

Wenn Ihre AWS CDK Bereitstellungsaktion mit einem `npm install` Fehler fehlschlägt, finden Sie unter Informationen [Wie behebe ich „npm install“-Fehler?](#) zur Behebung des Fehlers.

## YAML


So fügen Sie die Aktion „AWS CDK Bereitstellen“ mit dem YAML-Editor hinzu

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der AWS CDK Bereitstellungsaktion und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie AWS CDK Bereitstellen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld gehen Sie wie folgt vor:
  - (Optional) Wählen Sie „Herunterladen“, um [den Quellcode der Aktion anzuzeigen](#).
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „AWS CDK Bereitstellen“](#).
11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

 Note

Wenn Ihre AWS CDK Bereitstellungsaktion mit einem `npm install` Fehler fehlschlägt, finden Sie unter Informationen [Wie behebe ich „npm install“ -Fehler?](#) zur Behebung des Fehlers.

## Variablen, die durch die Aktion „AWS CDK Deploy“ erzeugt wurden

Wenn die AWS CDK Bereitstellungsaktion ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [AWS CDK Aktionsvariablen „bereitstellen“](#) in der [Liste der vordefinierten Variablen](#).

## Definition der Aktion „AWS CDK bereitstellen“

Die AWS CDK Bereitstellungsaktion ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „AWS CDK Bereitstellen“](#) in der [Referenz zur Workflow-Definition](#).

## Arbeiten mit Bereitstellungen

In diesem Abschnitt wird beschrieben, wie Bereitstellungen in Amazon CodeCatalyst verwaltet, überwacht und konfiguriert werden.

### Themen

- [Bereitstellung einer Anwendung oder Ressource](#)
- [Eine Bereitstellung erneut ausführen](#)
- [Bereitstellungsstatus, Commits und Pull-Requests anzeigen](#)
- [Bereitstellungsprotokolle anzeigen](#)

- [Rollbacks konfigurieren](#)
- [Anzeige der URL der bereitgestellten Anwendung](#)
- [Ein Bereitstellungsziel entfernen](#)

## Bereitstellung einer Anwendung oder Ressource

Um eine Anwendung oder Ressource mit Amazon bereitzustellen CodeCatalyst, müssen Sie einen CodeCatalyst Workflow verwenden. Dieser Workflow muss mindestens eine Aktion enthalten, mit der etwas bereitgestellt werden kann. Wenn Sie beispielsweise einen AWS CloudFormation Stack bereitstellen möchten, würden Sie einen Workflow erstellen, der die Aktion AWS CloudFormation Stack bereitstellen beinhaltet.

Weitere Informationen zu Workflows finden Sie unter [Erstellen, Testen und Bereitstellen mit Workflows in CodeCatalyst](#). Weitere Informationen zu Aktionen finden Sie unter [Verwenden von Aktionen](#).

## Eine Bereitstellung erneut ausführen

Um eine Bereitstellung in Amazon erneut auszuführen CodeCatalyst, führen Sie den zugehörigen Workflow erneut aus. Detaillierte Anweisungen finden Sie unter [Einen Workflow-Lauf starten](#).

## Bereitstellungsstatus, Commits und Pull-Requests anzeigen

Sie können die folgenden Informationen zu einer Bereitstellung in Amazon einsehen CodeCatalyst:

- Bereitstellungsaktivität, einschließlich Bereitstellungsstatus, Startzeit, Endzeit, Verlauf und Dauer der Ereignisse.
- Stackname AWS-Region, Uhrzeit der letzten Aktualisierung und zugehörige Workflows.
- Commits und Pull-Anfragen.
- Aktionsspezifische Informationen, zum Beispiel CloudFormation Ereignisse und Ausgaben.

[Sie können Bereitstellungsinformationen ausgehend von einem Workflow, einer Umgebung oder einer Workflow-Aktion anzeigen.](#)

So zeigen Sie Bereitstellungsinformationen ab einem Workflow an

- Gehen Sie zu der Workflow-Ausführung, die Ihre Anwendung bereitgestellt hat. Anweisungen finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).



## Um Bereitstellungsinformationen ausgehend von einer Umgebung anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Environments aus.
4. Wählen Sie die Umgebung aus, in der Ihr Stack bereitgestellt wurde, zum Beispiel. Production
5. Wählen Sie Bereitstellungsaktivität aus, um den Bereitstellungsverlauf Ihrer Stacks, den Status der Bereitstellungen (z. B. ERFOLGREICH ODER FEHLGESCHLAGEN) und andere Informationen zur Bereitstellung anzuzeigen.
6. Wählen Sie Bereitstellungsziel, um Informationen zu den Stacks, Clustern oder anderen Zielen anzuzeigen, die in der Umgebung bereitgestellt wurden. Sie können Informationen wie den Stacknamen, die Region, den Anbieter und die Kennung anzeigen.

## Um Bereitstellungsinformationen ab einer Aktion anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie im Workflow-Diagramm die Workflow-Aktion aus, mit der Ihre Anwendung bereitgestellt wurde. Sie könnten beispielsweise wählen DeployCloudFormationStack.
6. Im Inhalt des rechten Bereichs finden Sie aktionsspezifische Informationen zur Bereitstellung.

## Bereitstellungsprotokolle anzeigen

Sie können Protokolle zu bestimmten Bereitstellungsaktionen einsehen, um Probleme in Amazon zu beheben CodeCatalyst.

Sie können Protokolle ausgehend von einem [Workflow](#) oder einer [Umgebung](#) anzeigen.

## Um die Protokolle einer Bereitstellungsaktion ausgehend von einem Workflow anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.

3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Läufe aus.
6. Wählen Sie die Workflow-Ausführung aus, mit der Ihre Anwendung bereitgestellt wurde.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, deren Protokolle Sie anzeigen möchten.
8. Wählen Sie die Registerkarte Protokolle und erweitern Sie die Abschnitte, um die Protokollmeldungen anzuzeigen.
9. Um weitere Protokolle anzuzeigen, wählen Sie die Registerkarte Zusammenfassung und dann Ansicht in CloudFormation (falls verfügbar), um dort weitere Protokolle anzuzeigen. Möglicherweise müssen Sie sich anmelden bei AWS.

Um die Protokolle einer Bereitstellungsaktion ausgehend von einer Umgebung anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Environments aus.
4. Wählen Sie die Umgebung aus, in der Ihre Anwendung bereitgestellt wurde.
5. Suchen Sie unter Bereitstellungsaktivität nach der Spalte Workflow-Ausführungs-ID und wählen Sie den Workflow-Lauf aus, der Ihren Stack bereitgestellt hat.
6. Wählen Sie im Workflow-Diagramm die Aktion aus, deren Protokolle Sie anzeigen möchten.
7. Wählen Sie die Registerkarte Protokolle und erweitern Sie die Abschnitte, um die Protokollmeldungen anzuzeigen.
8. Um weitere Protokolle anzuzeigen, wählen Sie die Registerkarte Zusammenfassung und dann Ansicht in CloudFormation (falls verfügbar), um dort weitere Protokolle anzuzeigen. Möglicherweise müssen Sie sich anmelden bei AWS.


## Rollbacks konfigurieren

Wenn die Aktion `AWS CloudFormation Stack bereitstellen` fehlschlägt, führt dies standardmäßig AWS CloudFormation dazu, dass der Stack auf den letzten bekannten stabilen Status zurückgesetzt wird. Sie können das Verhalten so ändern, dass Rollbacks nicht nur dann auftreten, wenn die Aktion fehlschlägt, sondern auch, wenn ein bestimmter CloudWatch Amazon-Alarm ausgelöst wird. Weitere

Informationen zu CloudWatch Alarmen finden Sie unter [Verwenden von CloudWatch Amazon-Alarmen](#) im CloudWatch Amazon-Benutzerhandbuch.

Sie können auch das Standardverhalten so ändern, dass der Stack CloudFormation nicht zurückgesetzt wird, wenn die Aktion fehlschlägt.

Verwenden Sie die folgenden Anweisungen, um Rollbacks zu konfigurieren.

 Note

Sie können ein Rollback nicht manuell starten.

## Visual

Bevor Sie beginnen

1. Stellen Sie sicher, dass Sie über einen [Workflow](#) verfügen, der eine funktionierende Aktion „AWS CloudFormation Stack bereitstellen“ beinhaltet. Weitere Informationen finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#).
2. Stellen Sie sicher, dass Sie in der Rolle, die im Feld Stack-Rolle — optional der Aktion AWS CloudFormation Stack bereitstellen angegeben ist, die CloudWatchFullAccessBerechtigung angeben. Informationen zum Erstellen dieser Rolle mit den entsprechenden Berechtigungen finden Sie unter [Schritt 2: AWS Rollen erstellen](#).


So konfigurieren Sie Rollback-Alarme für die Aktion „AWS CloudFormation Stack bereitstellen“

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen eines Workflows, der die Aktion AWS CloudFormation Stack bereitstellen enthält. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie Ihre AWS CloudFormation Stack-Aktion Deploy aus.
8. Wählen Sie im Detailbereich Konfiguration aus.

9. Erweitern Sie unten die Option Erweitert.
10. Wählen Sie unter Alarm-ARNs überwachen die Option Alarm hinzufügen aus.
11. Geben Sie Informationen in die folgenden Felder ein.

- Alarm ARN

Geben Sie den Amazon-Ressourcennamen (ARN) eines CloudWatch Amazon-Alarms an, der als Rollback-Trigger verwendet werden soll. z. B. `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Sie können maximal fünf Rollback-Trigger haben.

 Note

Wenn Sie einen CloudWatch Alarm-ARN angeben, müssen Sie auch zusätzliche Berechtigungen konfigurieren, damit die Aktion darauf zugreifen kann CloudWatch. Weitere Informationen finden Sie unter [Rollbacks konfigurieren](#).

- Dauer der Überwachung

Geben Sie einen Zeitraum von 0 bis 180 Minuten an, in dem die angegebenen Alarme CloudFormation überwacht werden. Die Überwachung beginnt, nachdem alle Stack-Ressourcen bereitgestellt wurden. Wenn der Alarm innerhalb der angegebenen Überwachungszeit auftritt, CloudFormation schlägt die Bereitstellung fehl und der gesamte Stack-Vorgang wird rückgängig gemacht.

Standard: 0. CloudFormation überwacht Alarme nur während der Bereitstellung der Stack-Ressourcen, nicht danach.

## YAML

Um Rollback-Trigger für die Aktion „AWS CloudFormation Stack bereitstellen“ zu konfigurieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen eines Workflows, der die Aktion AWS CloudFormation Stack bereitstellen enthält. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.

5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Fügen Sie die `monitor-timeout-in-minutes` Eigenschaften `monitor-alarm-arns` und im YAML-Code hinzu, um Rollback-Trigger hinzuzufügen. Eine Erläuterung der einzelnen Eigenschaften finden Sie unter [Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“](#)
8. Stellen Sie sicher, dass Sie in der Rolle, die in der `role-arn` Eigenschaft der Aktion `AWS CloudFormation Stack bereitstellen` angegeben ist, die `CloudWatchFullAccess` Berechtigung angeben. Informationen zum Erstellen dieser Rolle mit den entsprechenden Berechtigungen finden Sie unter [Schritt 2: AWS Rollen erstellen](#).

## Visual

So deaktivieren Sie Rollbacks für die Aktion „AWS CloudFormation Stack bereitstellen“

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen eines Workflows, der die Aktion `AWS CloudFormation Stack bereitstellen` enthält. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie Ihre `AWS CloudFormation Stack-Aktion Deploy` aus.
8. Wählen Sie im Detailbereich Konfiguration aus.
9. Erweitern Sie unten die Option Erweitert.
10. Aktivieren Sie „Rollback deaktivieren“.

## YAML

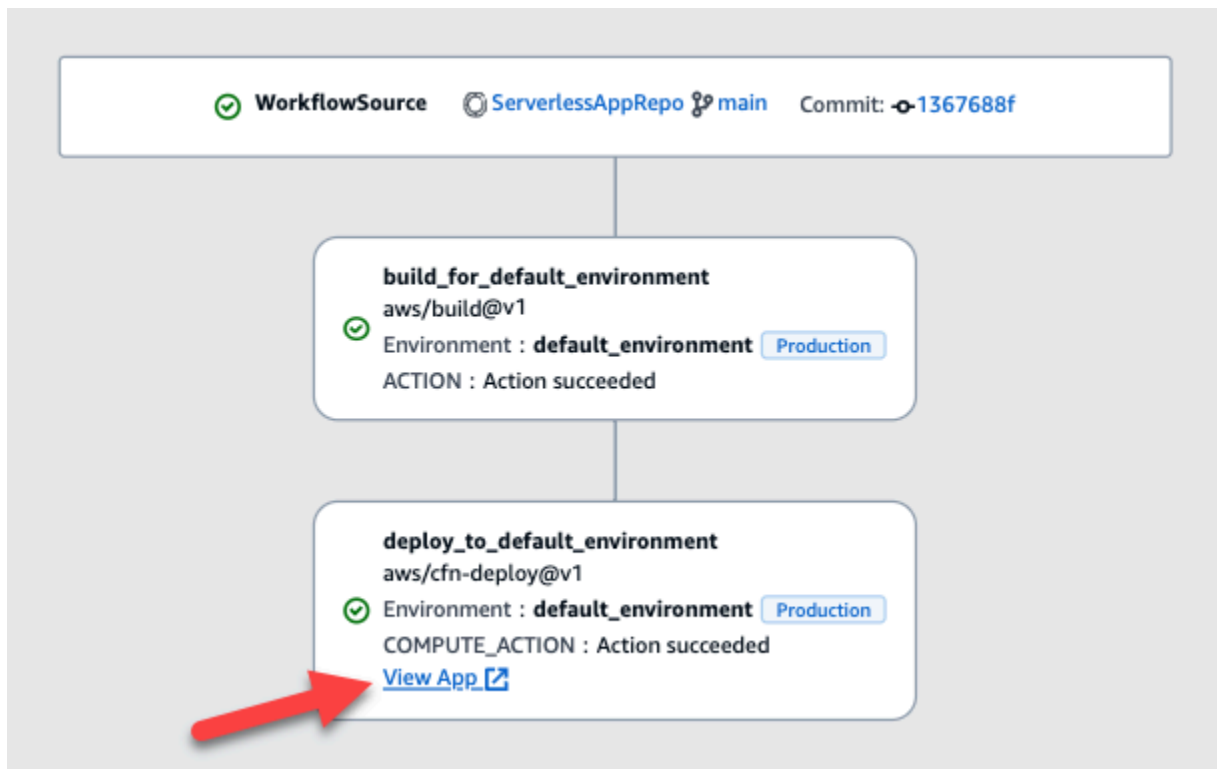
Um Rollbacks für die Aktion „AWS CloudFormation Stack bereitstellen“ zu deaktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.

3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen eines Workflows, der die Aktion `AWS CloudFormation Stack bereitstellen` enthält. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Fügen Sie die `disable-rollback: 1` Eigenschaft im YAML-Code hinzu, um Rollbacks zu verhindern. Eine Erläuterung dieser Eigenschaft finden Sie unter [Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“](#)

## Anzeige der URL der bereitgestellten Anwendung

Wenn Ihr Workflow eine Anwendung bereitstellt, können Sie Amazon so konfigurieren, CodeCatalyst dass die URL der Anwendung als anklickbarer Link angezeigt wird. Dieser Link wird in der CodeCatalyst Konsole in der Aktion angezeigt, mit der er bereitgestellt wurde. Das folgende Workflow-Diagramm zeigt die View-App-URL, die am Ende einer Aktion angezeigt wird.



Indem Sie diese URL in der CodeCatalyst Konsole anklickbar machen, können Sie Ihre Anwendungsbereitstellung schnell überprüfen.

**Note**

Die App-URL wird mit der Aktion Deploy to Amazon ECS nicht unterstützt.

Um diese Funktion zu aktivieren, fügen Sie Ihrer Aktion eine Ausgabevariable mit einem Namen hinzu `appurl`, der oder enthält `endpointurl`. Sie können einen Namen mit oder ohne verbindenden Bindestrich (-), Unterstrich ( \_ ) oder Leerzeichen ( ) verwenden. Bei der Zeichenfolge wird nicht zwischen Groß- und Kleinschreibung unterschieden. Setzen Sie den Wert der Variablen auf die `https` URL `http` oder die URL Ihrer bereitgestellten Anwendung.

**Note**

Wenn Sie eine bestehende Ausgabevariable so aktualisieren `app url`, dass sie die `endpoint url` Zeichenfolge oder die Zeichenfolge enthält, aktualisieren Sie alle Verweise auf diese Variable, sodass sie den neuen Variablennamen verwenden.

Ausführliche Schritte finden Sie in einem der folgenden Verfahren:

- [Um die App-URL in der Aktion „AWS CDK Bereitstellen“ anzuzeigen](#)
- [Um die App-URL in der Aktion „AWS CloudFormation Stack bereitstellen“ anzuzeigen](#)
- [Um die App-URL in allen anderen Aktionen anzuzeigen](#)

Wenn Sie mit der Konfiguration der URL fertig sind, stellen Sie sicher, dass sie wie erwartet aussieht, indem Sie die folgenden Anweisungen befolgen:

- [Um zu überprüfen, ob die Anwendungs-URL hinzugefügt wurde](#)

Um die App-URL in der Aktion „AWS CDK Bereitstellen“ anzuzeigen

1. Wenn Sie die Aktion „AWS CDK Bereitstellen“ verwenden, fügen Sie Ihrem AWS CDK Anwendungscode ein `CfnOutput` Konstrukt (ein Schlüssel-Wert-Paar) hinzu:
  - Der Schlüsselname muss oder mit oder `endpointurl` ohne einen verbindenden Bindestrich (-), Unterstrich ( \_ ) oder Leerzeichen ( ) enthalten `appurl`. Bei der Zeichenfolge wird nicht zwischen Groß- und Kleinschreibung unterschieden.

- Der Wert muss die `https` URL `http` oder der URL Ihrer bereitgestellten Anwendung sein.

Ihr AWS CDK Code könnte beispielsweise so aussehen:

```
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, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'APP-URL', {
      value: https://mycompany.myapp.com,
      description: 'The URL of the deployed application',
      exportName: 'myApp',
    });
    ...
  }
}
```

Weitere Informationen zu dem `CfnOutput` Konstrukt finden Sie unter [Interface CfnOutputProps](#) in der AWS Cloud Development Kit (AWS CDK) API-Referenz.

2. Speichern Sie Ihren Code und übertragen Sie ihn.
3. Fahren Sie mit [Um zu überprüfen, ob die Anwendungs-URL hinzugefügt wurde](#) fort.

Um die App-URL in der Aktion „AWS CloudFormation Stack bereitstellen“ anzuzeigen

1. Wenn Sie die Aktion `AWS CloudFormation Stack bereitstellen` verwenden, fügen Sie dem `Outputs` Abschnitt in Ihrer CloudFormation Vorlage oder AWS SAM Vorlage eine Ausgabe mit den folgenden Eigenschaften hinzu:
  - Der Schlüssel (auch logische ID genannt) muss oder mit oder `endpointurl` ohne einen verbindenden Bindestrich (`-`), Unterstrich (`_`) oder Leerzeichen () enthalten `appurl`. Bei der Zeichenfolge wird nicht zwischen Groß- und Kleinschreibung unterschieden.
  - Der Wert muss die `https` URL `http` oder der URL Ihrer bereitgestellten Anwendung sein.



Ihre CloudFormation Vorlage könnte beispielsweise so aussehen:

```
"Outputs" : {  
  "APP-URL" : {  
    "Description" : "The URL of the deployed app",  
    "Value" : "https://mycompany.myapp.com",  
    "Export" : {  
      "Name" : "My App"  
    }  
  }  
}
```

Weitere Informationen zu CloudFormation [Ausgaben](#) finden Sie im AWS CloudFormation Benutzerhandbuch unter Ausgaben.

2. Speichern Sie Ihren Code und geben Sie ihn ein.
3. Fahren Sie mit [Um zu überprüfen, ob die Anwendungs-URL hinzugefügt wurde](#) fort.

Um die App-URL in allen anderen Aktionen anzuzeigen

Wenn Sie eine andere Aktion verwenden, um Ihre Anwendung bereitzustellen, z. B. die Build-Aktion oder GitHub Aktionen, gehen Sie wie folgt vor, damit die App-URL angezeigt wird.

1. Definieren Sie eine Umgebungsvariable im Steps Abschnitt Inputs oder der Aktion in der Workflow-Definitionsdatei. Die Variable muss die folgenden Eigenschaften haben:
  - Die name muss oder appurlendpointurl, mit oder ohne einen verbindenden Bindestrich (-), Unterstrich (\_) oder Leerzeichen ( ) enthalten. Die Zeichenfolge unterscheidet nicht zwischen Groß- und Kleinschreibung.
  - Der Wert muss die https URL http oder der URL Ihrer bereitgestellten Anwendung sein.

Eine Build-Aktion könnte beispielsweise so aussehen:

```
Build-action:  
  Identifier: aws/build@v1  
  Inputs:  
  Variables:  
    - Name: APP-URL
```

```
Value: https://mycompany.myapp.com
```

... oder das:

```
Actions:  
  Build:  
    Identifier: aws/build@v1  
    Configuration:  
      Steps:  
        - Run: APP-URL=https://mycompany.myapp.com
```

Weitere Hinweise zur Definition von Umgebungsvariablen finden Sie unter [Definition einer Variablen](#).

2. Exportieren Sie die Variable.

Ihre Build-Aktion könnte beispielsweise so aussehen:

```
Build-action:  
  ...  
  Outputs:  
    Variables:  
      - APP-URL
```

Hinweise zum Exportieren von Variablen finden Sie unter [Eine Variable exportieren, sodass sie von anderen Aktionen verwendet werden kann](#).

3. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
4. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.
5. Fahren Sie mit [Um zu überprüfen, ob die Anwendungs-URL hinzugefügt wurde](#) fort.

Um zu überprüfen, ob die Anwendungs-URL hinzugefügt wurde

- Starten Sie eine Workflow-Ausführung, falls diese nicht automatisch gestartet wurde. Bei der neuen Ausführung sollte die App-URL als anklickbarer Link im Workflow-Diagramm angezeigt werden. Weitere Informationen zum Starten von Läufen finden Sie unter [Einen Workflow-Lauf starten](#).

## Ein Bereitstellungsziel entfernen

Sie können ein Bereitstellungsziel wie einen Amazon ECS-Cluster oder AWS CloudFormation -Stack von der Seite Umgebungen in der CodeCatalyst Konsole entfernen.

### Important

Wenn Sie ein Bereitstellungsziel entfernen, wird es aus der CodeCatalyst Konsole entfernt, bleibt aber in dem AWS Service verfügbar, der es hostet (sofern es noch existiert).

Erwägen Sie, ein Bereitstellungsziel zu entfernen, wenn das Ziel veraltet ist. CodeCatalyst Ziele könnten in folgenden Fällen veraltet sein:

- Sie haben den Workflow gelöscht, der für das Ziel bereitgestellt wurde.
- Sie haben den Stack oder Cluster geändert, auf dem Sie die Bereitstellung durchführen.
- Sie haben den Stack oder Cluster aus dem CloudFormation oder dem Amazon ECS-Service in der AWS Konsole gelöscht.

Um ein Bereitstellungsziel zu entfernen


1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Environments aus.
4. Wählen Sie den Namen der Umgebung aus, die das Bereitstellungsziel enthält, das Sie entfernen möchten. Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#).
5. Wählen Sie die Registerkarte Bereitstellungsziele.
6. Wählen Sie das Optionsfeld neben dem Bereitstellungsziel, das Sie entfernen möchten.
7. Wählen Sie Remove (Entfernen) aus.

Das Ziel wird von der Seite entfernt.

## Mit Workflows arbeiten

Ein Workflow ist ein automatisiertes Verfahren, das beschreibt, wie Sie Ihren Code als Teil eines CI/CD-Systems (Continuous Integration and Continuous Delivery) erstellen, testen und bereitstellen. Ein

Workflow definiert eine Reihe von Schritten oder Aktionen, die während einer Workflow-Ausführung ausgeführt werden sollen. Ein Workflow definiert auch die Ereignisse oder Auslöser, die dazu führen, dass der Workflow gestartet wird. Um einen Workflow einzurichten, erstellen Sie mit dem [visuellen Editor](#) oder dem [YAML-Editor](#) der CodeCatalyst Konsole eine Workflow-Definitionsdatei.

 Tip

Um einen schnellen Überblick darüber zu erhalten, wie Sie Workflows in einem Projekt verwenden könnten, [erstellen Sie ein Projekt mit einem Blueprint](#). Jeder Blueprint stellt einen funktionierenden Workflow bereit, den Sie überprüfen, ausführen und mit dem Sie experimentieren können.

## Themen

- [Einen Workflow erstellen, bearbeiten und löschen](#)
- [Anzeigen des Workflow-Status](#)
- [Verwenden von Aktionen](#)
- [Mit Artefakten arbeiten](#)
- [Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen](#)
- [Arbeiten mit Umgebungen](#)
- [Arbeiten mit Datei-Caching](#)
- [Mit Paketen arbeiten](#)
- [Mit Läufen arbeiten](#)
- [Arbeiten mit Secrets](#)
- [Mit Quellen arbeiten](#)
- [Arbeiten mit Auslösern](#)
- [Arbeiten mit Variablen](#)

## Einen Workflow erstellen, bearbeiten und löschen

Gehen Sie wie folgt vor, um einen Workflow zu erstellen, zu bearbeiten und zu löschen.

## Visual

Um einen Workflow mit dem Visual Editor zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie Workflow erstellen aus.


Das Dialogfeld „Workflow erstellen“ wird angezeigt.

5. Wählen Sie im Feld Quell-Repository ein Quell-Repository aus, in dem sich die Workflow-Definitionsdatei befinden soll. Die Datei wird in dem `~/codecatalyst/workflows/` Ordner im ausgewählten Repository gespeichert. Wenn kein Quell-Repository existiert, [erstellen Sie eines](#).
6. Wählen Sie im Feld Zweig einen Zweig aus, in dem sich die Workflow-Definitionsdatei befinden soll.
7. Wählen Sie Erstellen aus.

Amazon CodeCatalyst speichert das Repository und die Branch-Informationen im Speicher, aber der Workflow ist noch nicht festgeschrieben.

8. Wählen Sie Visual.
9. Erstellen Sie den Workflow:
  - a. (Optional) Wählen Sie im Workflow-Diagramm das Feld Quelle und Auslöser aus. Ein Bereich mit Auslösern wird angezeigt. Wählen Sie Trigger hinzufügen, um einen Trigger hinzuzufügen. Weitere Informationen finden Sie unter [Hinzufügen eines Push-, Pull- oder Zeitplanauslösers](#).
  - b. Wählen Sie + Aktionen (oben links). Der Aktionenkatalog wird angezeigt.
  - c. Wählen Sie das Pluszeichen (+) innerhalb einer Aktion, um sie dem Workflow hinzuzufügen. Verwenden Sie den Bereich auf der rechten Seite, um die Aktion zu konfigurieren. Weitere Informationen finden Sie unter [Hinzufügen einer Aktion](#).
  - d. (Optional) Wählen Sie Workflow-Eigenschaften (oben rechts). Ein Fenster mit den Workflow-Eigenschaften wird angezeigt. Konfigurieren Sie den Workflow-Namen, den Ausführungsmodus und die Berechnung. Weitere Informationen erhalten Sie unter [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#) und [Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen](#).

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie „Bestätigen“ und gehen Sie im Dialogfeld „Workflow bestätigen“ wie folgt vor:
  - a. Behalten Sie für Workflow-Dateiname den Standardnamen bei oder geben Sie einen eigenen ein.
  - b. Behalten Sie für Commit-Nachricht die Standardnachricht bei oder geben Sie Ihre eigene ein.
  - c. Wählen Sie für Repository und Branch das Quell-Repository und den Branch für die Workflow-Definitionsdatei aus. Diese Felder sollten auf das Repository und den Branch gesetzt werden, die Sie zuvor im Dialogfeld Workflow erstellen angegeben haben. Sie können das Repository und den Branch jetzt ändern, wenn Sie möchten.

 Note

Nachdem Sie Ihre Workflow-Definitionsdatei übernommen haben, kann sie keinem anderen Repository oder Branch zugeordnet werden. Wählen Sie sie daher sorgfältig aus.

- d. Wählen Sie Commit, um die Workflow-Definitionsdatei zu übernehmen.

## YAML

Um einen Workflow mit dem YAML-Editor zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie Workflow erstellen aus.

Das Dialogfeld „Workflow erstellen“ wird angezeigt.

5. Wählen Sie im Feld Quell-Repository ein Quell-Repository aus, in dem sich die Workflow-Definitionsdatei befinden soll. Die Datei wird in dem `~/codecatalyst/workflows/` Ordner im ausgewählten Repository gespeichert. Wenn kein Quell-Repository existiert, [erstellen Sie eines](#).

6. Wählen Sie im Feld **Zweig** einen Zweig aus, in dem sich die Workflow-Definitionsdatei befinden soll.
7. Wählen Sie **Erstellen** aus.

Amazon CodeCatalyst speichert das Repository und die Branch-Informationen im Speicher, aber der Workflow ist noch nicht festgeschrieben.

8. Wählen Sie **YAML**.
9. Erstellen Sie den Workflow:
  - a. (Optional) Fügen Sie dem YAML-Code einen Trigger hinzu. Weitere Informationen finden Sie unter [Hinzufügen eines Push-, Pull- oder Zeitplanauslösers](#).
  - b. Wählen Sie **+ Aktionen** (oben links). Der Aktionenkatalog wird angezeigt.
  - c. Wählen Sie das Pluszeichen (+) innerhalb einer Aktion, um sie dem Workflow hinzuzufügen. Verwenden Sie den Bereich auf der rechten Seite, um die Aktion zu konfigurieren. Weitere Informationen finden Sie unter [Hinzufügen einer Aktion](#).
  - d. (Optional) Wählen Sie **Workflow-Eigenschaften** (oben rechts). Ein Fenster mit den Workflow-Eigenschaften wird angezeigt. Konfigurieren Sie den Workflow-Namen, den Ausführungsmodus und die Berechnung. Weitere Informationen erhalten Sie unter [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#) und [Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen](#).
10. (Optional) Wählen Sie **„Validieren“**, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie **„Bestätigen“** und gehen Sie im Dialogfeld **„Workflow bestätigen“** wie folgt vor:
  - a. Behalten Sie für **Workflow-Dateiname** den Standardnamen bei oder geben Sie einen eigenen ein.
  - b. Behalten Sie für **Commit-Nachricht** die Standardnachricht bei oder geben Sie Ihre eigene ein.
  - c. Wählen Sie für **Repository** und **Branch** das Quell-Repository und den Branch für die Workflow-Definitionsdatei aus. Diese Felder sollten auf das Repository und den Branch gesetzt werden, die Sie zuvor im Dialogfeld **Workflow erstellen** angegeben haben. Sie können das Repository und den Branch jetzt ändern, wenn Sie möchten.

**Note**

Nachdem Sie Ihre Workflow-Definitionsdatei übernommen haben, kann sie keinem anderen Repository oder Branch zugeordnet werden. Wählen Sie sie daher sorgfältig aus.

- d. Wählen Sie Commit, um die Workflow-Definitionsdatei zu übernehmen.

### Um einen Workflow zu bearbeiten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen des Workflows aus, den Sie bearbeiten möchten. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Bearbeiten Sie den Workflow. Wählen Sie Visual, um den visuellen Editor zu verwenden, oder YAML, um den YAML-Editor zu verwenden. Sie haben folgende Möglichkeiten:
  - a. Trigger bearbeiten. Weitere Informationen finden Sie unter [Arbeiten mit Auslösern](#).
  - b. Aktionen bearbeiten. Weitere Informationen finden Sie unter [Verwenden von Aktionen](#).
7. Bearbeiten Sie zusätzliche Workflow-Eigenschaften. Wählen Sie Workflow-Eigenschaften (oben rechts), um den Workflow-Namen, den Ausführungsmodus oder die Berechnung zu ändern. Weitere Informationen erhalten Sie unter [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#) und [Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen](#).

### Um einen Workflow zu löschen

Beim Löschen eines Workflows werden die Workflow-Definitionsdatei und die zugehörigen Läufe gelöscht. Alle AWS Dienste, die aus dem gelöschten Workflow bereitgestellt wurden, müssen manuell entfernt werden, da sonst weiterhin Gebühren für diese Dienste anfallen.

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.



2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen des Workflows aus, den Sie löschen möchten. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Löschen.
6. Wählen Sie Bestätigen aus.

## Anzeigen des Workflow-Status

Möglicherweise möchten Sie den Status eines Workflows anzeigen, um festzustellen, ob Probleme mit der Workflow-Konfiguration behoben werden müssen, oder um Ausführungen zu beheben, die nicht gestartet werden können. CodeCatalyst wertet den Workflow-Status jedes Mal aus, wenn Sie die dem Workflow zugrunde [liegende Workflow-Definitionsdatei](#) erstellen oder aktualisieren.

### Note

Sie können auch den Ausführungsstatus des Workflows anzeigen, der sich vom Workflow-Status unterscheidet. Weitere Informationen finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).

Ein Workflow kann einen der folgenden Status haben:

- Gültig – Der Workflow ist ausführbar und kann durch [Auslöser](#) aktiviert werden.

Damit ein Workflow als gültig markiert wird, müssen die beiden folgenden Bedingungen erfüllt sein:

- Die Workflow-Definitionsdatei muss gültig sein.
- Der Workflow darf keine Auslöser, keine Push-Auslöser oder einen Push-Auslöser haben, der mit den Dateien auf dem aktuellen Zweig ausgeführt wird. Weitere Informationen finden Sie unter [Überlegungen zu Auslösern bei der Verzweigung](#).
- Nicht gültig – Die Definitionsdatei des Workflows ist ungültig. Der Workflow kann nicht manuell oder automatisch über Auslöser ausgeführt werden. Workflows, die nicht gültig sind, werden mit einer Workflow-Definition mit der Fehlermeldung **n** (oder ähnlichem) in der CodeCatalyst Konsole angezeigt.

Damit ein Workflow als ungültig markiert wird, muss die folgende Bedingung erfüllt sein:

- Die Workflow-Definitionsdatei muss falsch konfiguriert sein.

Informationen zum Beheben einer falsch konfigurierten Workflow-Definitionsdatei finden Sie unter [Wie behebe ich den Fehler „Workflow-Definition enthält keine Fehler“?](#)

- Inaktiv – Die Workflow-Definition ist gültig, kann aber nicht manuell oder automatisch über Auslöser ausgeführt werden.

Damit ein Workflow als inaktiv markiert wird, müssen die beiden folgenden Bedingungen erfüllt sein:

- Die Workflow-Definitionsdatei muss gültig sein.
- Die Workflow-Definitionsdatei muss einen Push-Auslöser enthalten, der eine Verzweigung angibt, die sich von der unterscheidet, in der sich die Workflow-Definitionsdatei derzeit befindet. Weitere Informationen finden Sie unter [Überlegungen zu Auslösern bei der Verzweigung](#).

Informationen zum Wechseln eines Workflows von Inaktiv zu Aktiv finden Sie unter [Wie behebe ich die Meldung „Der Workflow ist inaktiv“?](#)

#### Note

Wenn der Workflow eine Ressource angibt, die Sie später entfernen (z. B. ein Paket-Repository), CodeCatalyst erkennt diese Änderung nicht und markiert den Workflow weiterhin als gültig. Diese Arten von Problemen werden erkannt, wenn der Workflow ausgeführt wird.

So zeigen Sie den Status eines Workflows an

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Suchen Sie den Workflow, dessen Status Sie anzeigen möchten. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.

Der Status wird mit dem Workflow in der Liste angezeigt.

5. (Optional) Wählen Sie den Namen des Workflows aus und suchen Sie das Feld Workflow-Definition. Es zeigt den Workflow-Status an.

## Verwenden von Aktionen

Eine Aktion ist der Hauptbaustein eines Workflows und definiert eine logische Arbeitseinheit, die während einer Workflow-Ausführung ausgeführt werden soll. In der Regel umfasst ein Workflow mehrere Aktionen, die nacheinander oder parallel ausgeführt werden, je nachdem, wie Sie sie konfiguriert haben.

### Themen

- [Aktionstypen](#)
- [Aktion hinzufügen, konfigurieren und entfernen](#)
- [Entwicklung einer benutzerdefinierten Aktion](#)
- [Gruppierung von Aktionen in Aktionsgruppen](#)
- [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#)
- [Den Quellcode einer Aktion anzeigen](#)
- [Mit Action-Versionen arbeiten](#)

## Aktionstypen

Innerhalb eines Amazon- CodeCatalyst Workflows können Sie die folgenden Arten von Aktionen verwenden.

### Aktionstypen

- [CodeCatalyst -Aktionen](#)
- [CodeCatalyst Labs-Aktionen](#)
- [GitHub Aktionen](#)
- [Drittanbieteraktionen](#)

### CodeCatalyst -Aktionen

Eine CodeCatalyst Aktion ist eine Aktion, die vom CodeCatalyst Entwicklungsteam verfasst, verwaltet und vollständig unterstützt wird.

Es gibt CodeCatalyst Aktionen zum Erstellen, Testen und Bereitstellen von Anwendungen sowie zum Ausführen verschiedener Aufgaben, wie z. B. das Aufrufen einer - AWS Lambda Funktion.

Die folgenden CodeCatalyst Aktionen sind verfügbar:

- Entwicklung

Diese Aktion erstellt Ihre Artefakte und führt Ihre Einheitentests in einem Docker-Container aus. Weitere Informationen finden Sie unter [Hinzufügen der Build-Aktion](#).

- Test

Diese Aktion führt Integrations- und Systemtests für Ihre Anwendung oder Artefakte durch. Weitere Informationen finden Sie unter [Testaktion hinzufügen](#).

- Amazon S3 veröffentlichen

Diese Aktion kopiert Ihre Anwendungsartefakte in einen Amazon S3-Bucket. Weitere Informationen finden Sie unter [Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen](#).

- AWS CDK Bootstrap

Diese Aktion stellt die Ressourcen bereit, die der AWS CDK benötigt, um Ihre CDK-App bereitzustellen. Weitere Informationen finden Sie unter [Aktion „AWS CDK Bootstrap“ hinzufügen](#).

- AWS CDK Bereitstellen

Diese Aktion synthetisiert und stellt eine AWS Cloud Development Kit (AWS CDK) App bereit. Weitere Informationen finden Sie unter [Aktion „AWS CDK Deploy“ hinzufügen](#).

- AWS Lambda aufrufen

Diese Aktion ruft eine - AWS Lambda Funktion auf. Weitere Informationen finden Sie unter [Hinzufügen der Aktion „AWS LambdaAufrufen“](#).

- Bereitstellen eines AWS CloudFormation Stacks

Diese Aktion stellt AWS CloudFormation Stacks bereit. Weitere Informationen finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#).

- Bereitstellen auf Amazon ECS

Diese Aktion registriert eine Amazon-ECS-Aufgabendefinition und stellt sie für einen Amazon-ECS-Service bereit. Weitere Informationen finden Sie unter [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#).

- Bereitstellen im Kubernetes-Cluster

Diese Aktion stellt eine Anwendung in einem Kubernetes-Cluster bereit. Weitere Informationen finden Sie unter [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#).

- Rendern der Amazon-ECS-Aufgabendefinition

Diese Aktion fügt einen Container-Image-URI in eine JSON-Datei der Amazon-ECS-Aufgabendefinition ein und erstellt eine neue Aufgabendefinitionsdatei. Weitere Informationen finden Sie unter [Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen](#).

Die Dokumentation für - CodeCatalyst Aktionen ist in diesem Handbuch und in der Readme jeder Aktion verfügbar.

Informationen zu den verfügbaren CodeCatalyst Aktionen und zum Hinzufügen einer Aktion zu einem Workflow finden Sie unter [Hinzufügen einer Aktion](#).

### CodeCatalyst Labs-Aktionen

Eine CodeCatalyst Labs-Aktion ist eine Aktion, die Teil von Amazon CodeCatalyst Labs ist, einem Beweisgrund für experimentelle Anwendungen. CodeCatalyst Labs-Aktionen wurden entwickelt, um Integrationen mit - AWS Services zu demonstrieren.

Die folgenden CodeCatalyst Labs-Aktionen sind verfügbar:

- Bereitstellen auf AWS Amplify Hosting

Diese Aktion stellt eine Anwendung für Amplify Hosting bereit.

- Bereitstellen in AWS App Runner

Diese Aktion stellt das neueste Image in einem Quell-Image-Repository in App Runner bereit.

- Bereitstellen in Amazon CloudFront und Amazon S3

Diese Aktion stellt eine Anwendung in CloudFront und Amazon S3 bereit.

- Bereitstellen mit AWS SAM

Diese Aktion stellt Ihre Serverless-Anwendung mit AWS Serverless Application Model (AWS SAM) bereit.

- Aufheben der Gültigkeit von Amazon CloudFront Cache

Diese Aktion macht einen CloudFront Cache für einen bestimmten Satz von Pfaden ungültig.

- Ausgehende Webhook-Aktion

Diese Aktion ermöglicht es Benutzern, Nachrichten innerhalb eines Workflows mithilfe einer HTTPS-Anforderung an einen beliebigen Webserver zu senden.

- Veröffentlichen in AWS CodeArtifact

Diese Aktion veröffentlicht Pakete in einem CodeArtifact Repository.

- In Amazon SNS veröffentlichen

Diese Aktion ermöglicht es Benutzern, sich in Amazon SNS zu integrieren, indem sie ein Thema erstellen, ein Thema veröffentlichen oder ein Thema abonnieren.

- Push zu Amazon ECR

Diese Aktion erstellt ein Docker-Image und veröffentlicht es in einem Amazon Elastic Container Registry (Amazon ECR)-Repository.

- Scannen mit Amazon CodeGuru Security

Diese Aktion erstellt ein ZIP-Archiv eines konfigurierten Codepfads und verwendet CodeGuru Sicherheit, um einen Code-Scan auszuführen.

- Terraform Community Edition

Diese Aktion führt die apply Operationen Terraform Community Edition plan und aus.

Die Dokumentation für CodeCatalyst Labs-Aktionen ist in der Readme jeder Aktion verfügbar.

Informationen zum Hinzufügen einer CodeCatalyst Labs-Aktion zu einem Workflow und zum Anzeigen seiner Readme finden Sie unter [Hinzufügen einer Aktion](#).

## GitHub Aktionen

Eine GitHub Aktion ist einer [CodeCatalyst Aktion](#) sehr ähnlich, mit der Ausnahme, dass sie für die Verwendung mit GitHub Workflows entwickelt wurde. Weitere Informationen zu GitHub Aktionen finden Sie in der Dokumentation zu [GitHub Aktionen](#).

Sie können eine GitHub Aktion zusammen mit nativen CodeCatalyst Aktionen in einem CodeCatalyst Workflow verwenden.

Der Einfachheit halber bietet die CodeCatalyst Konsole Zugriff auf mehrere beliebige GitHub Aktionen. Sie können auch jede im [GitHub Marketplace](#) aufgeführte GitHub Aktion verwenden (vorbehaltlich einiger Einschränkungen).

Die Dokumentation für GitHub Aktionen ist in der Readme jeder Aktion verfügbar.

Weitere Informationen finden Sie unter [GitHub Aktionen hinzufügen](#).

## Drittanbieteraktionen

Eine Aktion eines Drittanbieters ist eine Aktion, die von einem Drittanbieter erstellt und in der CodeCatalyst Konsole verfügbar gemacht wird. Ein Beispiel für eine Drittanbieteraktion ist die Mend-SCA-Aktion, die von Mend verfasst wurde.

Die Dokumentation für Aktionen von Drittanbietern ist in der Readme jeder Aktion verfügbar. Zusätzliche Dokumentation kann auch vom Drittanbieter bereitgestellt werden.

Informationen zum Hinzufügen einer Drittanbieteraktion zu einem Workflow und zum Anzeigen der Readme finden Sie unter [Hinzufügen einer Aktion](#).

## Aktion hinzufügen, konfigurieren und entfernen

Wählen Sie eines der folgenden Themen aus, um zu erfahren, wie Sie die Aktion zu einem Workflow hinzufügen und wie Sie sie entfernen können.

### Themen

- [Hinzufügen einer Aktion](#)
- [Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen](#)
- [Aktion „AWS CDK Bootstrap“ hinzufügen](#)
- [Hinzufügen der Aktion „AWS LambdaAufrufen“](#)
- [Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen](#)
- [GitHub Aktionen hinzufügen](#)
- [Eine Aktion entfernen](#)

### Hinzufügen einer Aktion

Verwenden Sie die folgenden Anweisungen, um einem Workflow eine Aktion hinzuzufügen und sie dann zu konfigurieren.

So fügen Sie eine Aktion hinzu und konfigurieren sie

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie oben links + Aktionen , der Katalog Aktionen wird angezeigt.
7. Führen Sie in der Dropdown-Liste einen der folgenden Schritte aus:
  - Wählen Sie Amazon CodeCatalyst aus, um [CodeCatalyst](#), [CodeCatalyst Labs](#) oder Aktionen [von Drittanbietern](#) anzuzeigen.
    - CodeCatalyst -Aktionen haben eine nach AWS Label.
    - CodeCatalyst Labs-Aktionen haben eine nach CodeCatalyst Labs-Bezeichnung.
    - Aktionen von Drittanbietern haben eine nach **Anbieterbezeichnung**, wobei **Anbieter** der Name des Drittanbieters ist.
  - Wählen Sie GitHub, um eine [kuratierte Liste der GitHub Aktionen](#) anzuzeigen.
8. Suchen Sie im Aktionskatalog nach einer Aktion und führen Sie dann einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zu Ihrem Workflow hinzuzufügen.
  - Wählen Sie den Namen der Aktion aus, um deren Readme anzuzeigen.
9. Konfigurieren Sie die Aktion . Wählen Sie Visual, um den visuellen Editor zu verwenden, oder YAML, um den YAML-Editor zu verwenden. Detaillierte Anweisungen finden Sie unter den folgenden Links.

Anweisungen zum Hinzufügen von [CodeCatalyst Aktionen](#) finden Sie unter:

- [Hinzufügen der Build-Aktion](#)
- [Testaktion hinzufügen](#)
- [Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen](#)
- [Aktion „AWS CDK Bootstrap“ hinzufügen](#)
- [Aktion „AWS CDK Deploy“ hinzufügen](#)



- [Hinzufügen der Aktion „AWS LambdaAufrufen“](#)
- [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#)
- [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#)
- [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#)
- [Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen](#)

Anweisungen zum Hinzufügen von [CodeCatalyst -Labs-Aktionen](#) finden Sie unter:

- Die Readme der Aktion. Sie finden das Readme, indem Sie den Namen der Aktion im Aktionskatalog auswählen.

Anweisungen zum Hinzufügen von [GitHub Aktionen](#) finden Sie unter:

- [GitHub Aktionen hinzufügen](#)

Anweisungen zum Hinzufügen [von Aktionen von Drittanbietern](#) finden Sie unter:

- Die Readme der Aktion. Sie finden das Readme, indem Sie den Namen der Aktion im Aktionskatalog auswählen.

10. (Optional) Wählen Sie Validieren, um sicherzustellen, dass der YAML-Code gültig ist.

11. Wählen Sie Commit aus, um Ihre Änderungen zu bestätigen.

Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen

In diesem Abschnitt wird beschrieben, wie Sie die Amazon S3 S3-Aktion „Veröffentlichen“ zu Ihrem Workflow hinzufügen. Die Amazon S3 S3-Veröffentlichungsaktion kopiert Dateien aus einem Quellverzeichnis in einen Amazon S3 S3-Bucket. Das Quellverzeichnis kann sich in folgendem Verzeichnis befinden:

- Ein [Quell-Repository](#) oder
- Ein [Ausgabeartefakt](#), das durch eine andere Workflow-Aktion generiert wurde

Themen

- [Wann sollte diese Aktion verwendet werden](#)
- [Beispiel für einen Arbeitsablauf](#)

- [Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen](#)
- [Variablen, die durch die Aktion „Amazon S3 Publish“ erzeugt wurden](#)
- [Aktionsdefinition „Amazon S3 Publish“](#)

Wann sollte diese Aktion verwendet werden

Verwenden Sie diese Aktion, wenn:

- Sie haben einen Workflow, der Dateien generiert, die Sie in Amazon S3 speichern möchten.

Möglicherweise haben Sie einen Workflow, der eine statische Website erstellt, die Sie in Amazon S3 hosten möchten. In diesem Fall würde Ihr Workflow eine [Build-Aktion](#) zum Erstellen der HTML- und unterstützenden Dateien der Site sowie eine Amazon S3-Veröffentlichungsaktion zum Kopieren der Dateien nach Amazon S3 beinhalten.

- Sie haben ein Quell-Repository, das Dateien enthält, die Sie in Amazon S3 speichern möchten.

Möglicherweise haben Sie ein Quell-Repository mit Anwendungsquelldateien, die Sie jede Nacht in Amazon S3 archivieren möchten.

Beispiel für einen Arbeitsablauf

Der folgende Beispiel-Workflow umfasst die Amazon S3 S3-Veröffentlichungsaktion zusammen mit einer Build-Aktion. Der Workflow erstellt eine statische Dokumentationswebsite und veröffentlicht sie dann in Amazon S3, wo sie gehostet wird. Der Workflow besteht aus den folgenden Bausteinen, die nacheinander ausgeführt werden:

#### Note

Das folgende Workflow-Beispiel dient der Veranschaulichung und funktioniert nur mit zusätzlicher Konfiguration.

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine Build-Aktion (BuildDocs) — Beim Trigger erstellt die Aktion eine statische Dokumentationswebsite (`mkdocs build`) und fügt die zugehörigen HTML-Dateien und

unterstützenden Metadaten zu einem Artefakt mit dem Namen MyDocsSite hinzu. Weitere Informationen zur Build-Aktion finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#).

- Eine Amazon S3-Veröffentlichungsaktion (PublishToS3) — Nach Abschluss der Build-Aktion kopiert diese Aktion die Site im MyDocsSite Artefakt zum Hosten nach Amazon S3.

```
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:
```

```
- MyDocSite
Configuration:
  DestinationBucketName: my-bucket
  SourcePath: /artifacts/PublishToS3/MyDocSite/site
  TargetPath: my/docs/site
```

## Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen

Verwenden Sie die folgenden Anweisungen, um die Amazon S3 S3-Veröffentlichungsaktion zu Ihrem Workflow hinzuzufügen.

### Visual

Um die Aktion „Amazon S3 Publish“ mit dem Visual Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Edit (Bearbeiten) aus.
  6. Wählen Sie Visual.
  7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
  9. Suchen Sie nach der Amazon S3 S3-Aktion „Veröffentlichen“ und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Or
- Wählen Sie Amazon S3 Publish. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

10. Füllen Sie auf den Registerkarten Eingaben, Konfiguration und Ausgaben die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „Amazon S3 veröffentlichen“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

## YAML

Um die Aktion „Amazon S3 Publish“ mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Edit (Bearbeiten) aus.
  6. Wählen Sie YAML.
  7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
  9. Suchen Sie nach der Amazon S3 S3-Aktion „Veröffentlichen“ und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Or
- Wählen Sie Amazon S3 Publish. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.

- Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „Amazon S3 veröffentlichen“](#).
  11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
  12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

Variablen, die durch die Aktion „Amazon S3 Publish“ erzeugt wurden

Wenn die Amazon S3 S3-Veröffentlichungsaktion ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [Aktionsvariablen „Amazon S3 Publish“](#) in der [Liste der vordefinierten Variablen](#).

Aktionsdefinition „Amazon S3 Publish“

Die Amazon S3 S3-Veröffentlichungsaktion ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „Amazon S3 veröffentlichen“](#) in der [Referenz zur Workflow-Definition](#).

Aktion „AWS CDK Bootstrap“ hinzufügen

In diesem Abschnitt wird beschrieben, wie Sie die AWS CDK Bootstrap-Aktion zu Ihrem Workflow hinzufügen. [Die AWS CDK Bootstrap-Aktion stellt mithilfe der modernen Vorlage einen Bootstrap-Stack in Ihrer AWS Umgebung bereit](#). Wenn bereits ein Bootstrap-Stack vorhanden ist, aktualisiert die Aktion ihn bei Bedarf. Das Vorhandensein eines Bootstrap-Stacks AWS ist eine Voraussetzung für die Bereitstellung einer AWS CDK App.

Weitere Informationen zu Bootstrapping finden Sie unter [Bootstrapping](#) im Entwicklerhandbuch.AWS Cloud Development Kit (AWS CDK)

Themen

- [Wann sollte diese Aktion verwendet werden](#)
- [Funktionsweise](#)
- [CDK-CLI-Versionen, die von der Aktion „AWS CDK Bootstrap“ verwendet werden](#)

- [Voraussetzungen](#)
- [Beispiel-ArbeitsFlows](#)
- [Die Aktion „AWS CDK Bootstrap“ wird hinzugefügt](#)
- [Variablen, die durch die Aktion „AWS CDK Bootstrap“ erzeugt wurden](#)
- [Definition der Aktion „AWS CDK Bootstrap“](#)

Wann sollte diese Aktion verwendet werden

Verwenden Sie diese Aktion, wenn Sie über einen Workflow verfügen, der eine AWS CDK App bereitstellt, und Sie gleichzeitig den Bootstrap-Stack bereitstellen (und bei Bedarf aktualisieren) möchten. In diesem Fall würden Sie die AWS CDK Bootstrap-Aktion demselben Workflow hinzufügen wie dem, der Ihre App bereitstellt. AWS CDK

Verwenden Sie diese Aktion nicht, wenn eine der folgenden Bedingungen zutrifft:

- Sie haben bereits einen Bootstrap-Stack mit einem anderen Mechanismus bereitgestellt und möchten ihn beibehalten (keine Updates).
- Sie möchten eine [benutzerdefinierte Bootstrap-Vorlage verwenden, die von der AWS CDK Bootstrap-Aktion](#) nicht unterstützt wird.

Funktionsweise

Der AWS CDK Bootstrap funktioniert wie folgt:

1. [Wenn Sie zur Laufzeit Version 1.0.7 oder früher der Aktion angegeben haben, lädt die Aktion die neueste CDK-CLI \(auch AWS CDK Toolkit genannt\) in das Build-Image herunter. CodeCatalyst](#)

Wenn Sie Version 1.0.8 oder höher angegeben haben, ist die Aktion mit einer [bestimmten Version](#) der CDK-CLI gebündelt, sodass kein Download stattfindet.

2. Die Aktion verwendet die CDK-CLI, um den `cdk bootstrap` Befehl auszuführen. Mit diesem Befehl werden die Bootstrapping-Aufgaben ausgeführt, die im Thema [Bootstrapping im Entwicklerhandbuch](#) beschrieben sind. AWS Cloud Development Kit (AWS CDK)

CDK-CLI-Versionen, die von der Aktion „AWS CDK Bootstrap“ verwendet werden

Die folgende Tabelle zeigt, welche Version der CDK-CLI standardmäßig von verschiedenen Versionen der AWS CDK Bootstrap-Aktion verwendet wird.

**Note**

Möglicherweise können Sie die Standardeinstellung überschreiben. Weitere Informationen finden Sie unter [CdkCliVersion](#) im [Referenz zur Aktion „AWS CDK bootstrap“](#).

AWS CDK Aktionsversion „Bootstrap“	AWS CDK CLI-Version
1.0.0 — 1.0.7	brandneue
1.0.8 oder später	2.99.1

### Voraussetzungen

Bevor Sie die AWS CDK Bootstrap-Aktion verwenden können, stellen Sie sicher, dass Sie eine AWS CDK App bereit haben. Die Bootstrap-Aktion synthetisiert die AWS CDK App vor dem Bootstrapping. Sie können Ihre App in jeder Programmiersprache schreiben, die von der unterstützt wird. AWS CDK

Stellen Sie sicher, dass Ihre AWS CDK App-Dateien verfügbar sind in:

- Ein CodeCatalyst [Quell-Repository](#) oder
- Ein CodeCatalyst [Ausgabeartefakt](#), das durch eine andere Workflow-Aktion generiert wurde

### Beispiel-ArbeitsFlows

Einen Workflow, [Beispiel-ArbeitsFlows](#) der die [Aktion „AWS CDK Deploy“ hinzufügen](#) AWS CDK Bootstrap-Aktion beinhaltet, finden Sie im.

Die Aktion „AWS CDK Bootstrap“ wird hinzugefügt

Verwenden Sie die folgenden Anweisungen, um die AWS CDK Bootstrap-Aktion zu Ihrem Workflow hinzuzufügen.

### Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die unter beschriebenen Aufgaben abgeschlossen haben. [Voraussetzungen](#)



## Visual

Um die Aktion „AWS CDK Bootstrap“ mit dem visuellen Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der AWS CDK Bootstrap-Aktion und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie AWS CDK Bootstrap. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Füllen Sie auf den Registerkarten Eingaben, Konfiguration und Ausgaben die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „AWS CDK bootstrap“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
  11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
  12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

**Note**

Wenn Ihre AWS CDK Bootstrap-Aktion mit einem `npm install` Fehler fehlschlägt, finden Sie weitere Informationen [Wie behebe ich „npm install“-Fehler?](#) zur Behebung des Fehlers unter.

## YAML

So fügen Sie die Aktion „AWS CDK Bootstrap“ mit dem YAML-Editor hinzu

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Wählen Sie links oben + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der AWS CDK Bootstrap-Aktion und wählen Sie +, um sie dem Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „AWS CDK bootstrap“](#).
11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

**Note**

Wenn Ihre AWS CDK Bootstrap-Aktion mit einem `npm install` Fehler fehlschlägt, finden Sie weitere Informationen [Wie behebe ich „npm install“-Fehler?](#) zur Behebung des Fehlers unter.

Variablen, die durch die Aktion „AWS CDK Bootstrap“ erzeugt wurden

Wenn die AWS CDK Bootstrap-Aktion ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [AWS CDK Aktionsvariablen „Bootstrap“](#) in der [Liste der vordefinierten Variablen](#).

Definition der Aktion „AWS CDK Bootstrap“

Die AWS CDK Bootstrap-Aktion ist als ein Satz von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „AWS CDK bootstrap“](#) in der [Referenz zur Workflow-Definition](#)

Hinzufügen der Aktion „AWS Lambda Aufrufen“

In diesem Abschnitt wird beschrieben, wie Sie die AWS Lambda Aufrufaktion zu Ihrem Workflow hinzufügen. Die AWS Lambda Aufrufaktion ruft eine von Ihnen angegebene Lambda-Funktion auf.

Zusätzlich zum Aufrufen Ihrer Funktion konvertiert die AWS Lambda Aufrufaktion auch jeden Schlüssel der obersten Ebene in der von der Lambda-Funktion empfangenen Antwortnutzlast in eine [Workflow-Ausgabevariable](#). Diese Variablen können dann in nachfolgenden Workflow-Aktionen referenziert werden. Wenn nicht alle Schlüssel der obersten Ebene in Variablen konvertiert werden sollen, können Sie Filter verwenden, um die genauen Schlüssel anzugeben. Weitere Informationen finden Sie unter [-ResponseFilters](#) Eigenschaftsbeschreibung im [„AWS Lambda Aufruf“-Aktionsreferenz](#).

Themen

- [Wann sollte diese Aktion verwendet werden?](#)
- [Beispiel-ArbeitsFlows](#)
- [Hinzufügen der Aktion „AWS Lambda Aufrufen“](#)
- [Von der Aktion „AWS Lambda Aufruf“ erzeugte Variablen](#)

- ["AWS Lambda invoke"-Aktionsdefinition](#)


Wann sollte diese Aktion verwendet werden?

Verwenden Sie diese Aktion, wenn Sie Ihrem Workflow Funktionen hinzufügen möchten, die in einer Lambda-Funktion gekapselt und von dieser ausgeführt werden.

Beispielsweise möchten Sie vielleicht, dass Ihr Workflow eine `Build started` Benachrichtigung an einen Slack-Kanal sendet, bevor Sie einen Build Ihrer Anwendung starten. In diesem Fall enthält Ihr Workflow eine AWS Lambda Aufrufaktion zum Aufrufen eines Lambda zum Senden der Slack-Benachrichtigung und eine [Build-Aktion](#) zum Erstellen Ihrer Anwendung.

Ein weiteres Beispiel: Möglicherweise möchten Sie, dass Ihr Workflow einen Schwachstellenscan für Ihre Anwendung durchführt, bevor sie bereitgestellt wird. In diesem Fall würden Sie eine Build-Aktion verwenden, um Ihre Anwendung zu erstellen, eine AWS Lambda Aufruf-Aktion, um ein Lambda aufzurufen, um nach Schwachstellen zu suchen, und eine Bereitstellungsaktion, um die gescannte Anwendung bereitzustellen.

Beispiel-ArbeitsFlows

 Note

Der folgende Beispiel-Workflow dient nur zur Veranschaulichung und erfordert zusätzliche Einrichtungsarbeit, um ordnungsgemäß zu funktionieren. Es soll Ihnen ein Beispiel dafür geben, wie ein Workflow aussehen könnte, wenn er mit der AWS Lambda Aufrufaktion konfiguriert ist.

Der folgende Workflow enthält die AWS Lambda Aufrufaktion zusammen mit einer Bereitstellungsaktion. Der Workflow sendet eine Slack-Benachrichtigung, die angibt, dass eine Bereitstellung gestartet wurde, und stellt dann AWS mithilfe einer -AWS CloudFormation Vorlage eine Anwendung in bereit. Der Workflow besteht aus den folgenden Bausteinen, die sequenziell ausgeführt werden:

- Ein Auslöser – Dieser Auslöser startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übergeben. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine AWS Lambda Aufrufaktion (`LambdaNotify`) – Beim Auslöser ruft diese Aktion die `Notify-Start` Lambda-Funktion im angegebenen AWS Konto und in der angegebenen Region (`my-`

aws-account, und ) aufus-west-2. Beim Aufruf sendet die Lambda-Funktion eine Slack-Benachrichtigung, die angibt, dass eine Bereitstellung gestartet wurde.

- Eine AWS CloudFormationStack-Aktion bereitstellen (Deploy) – Nach Abschluss der AWS Lambda Aufrufaktion führt die Aktion AWS CloudFormationStack bereitstellen die Vorlage (cfn-template.yml) aus, um Ihren Anwendungs-Stack bereitzustellen. Weitere Informationen zur Aktion AWS CloudFormationStack bereitstellen finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#).

```
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
```

## Hinzufügen der Aktion „AWS LambdaAufrufen“

Verwenden Sie die folgenden Anweisungen, um die AWS Lambda Aufrufaktion zu Ihrem Workflow hinzuzufügen.

### Voraussetzung

Bevor Sie beginnen, stellen Sie sicher, dass Ihre AWS Lambda Funktion und die zugehörige Lambda-Ausführungsrolle bereit und in verfügbar sindAWS. Weitere Informationen finden Sie im Thema [Lambda-Ausführungsrolle](#) im AWS Lambda -Entwicklerhandbuch.

### Visual

So fügen Sie die Aktion „AWS LambdaAufruf“ mit dem visuellen Editor hinzu

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual aus.
7. Wählen Sie oben links + Aktionen aus, um den Aktionskatalog zu öffnen.
8. Wählen Sie aus der Dropdown-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der AWS Lambda Aufrufaktion und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie AWS Lambda Aufrufen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
  - (Optional) Wählen Sie Quelle anzeigen, um [den Quellcode der Aktion anzuzeigen](#).
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den Konfigurationsbereich zu öffnen.
- 10. Füllen Sie auf den Registerkarten Eingaben , Konfiguration und Ausgaben die Felder entsprechend Ihren Anforderungen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [„AWS Lambda Aufruf“-Aktionsreferenz](#). Diese Referenz enthält detaillierte Informationen über jedes Feld (und den entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
- 11. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
- 12. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

So fügen Sie die Aktion „AWS LambdaAufruf“ mit dem YAML-Editor hinzu

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML aus.
7. Wählen Sie oben links + Aktionen aus, um den Aktionskatalog zu öffnen.
8. Wählen Sie aus der Dropdown-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der AWS Lambda Aufrufaktion und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den Konfigurationsbereich zu öffnen.

Oder

- Wählen Sie AWS Lambda Aufrufen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
  - (Optional) Wählen Sie Quelle anzeigen, um [den Quellcode der Aktion anzuzeigen](#).
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den Konfigurationsbereich zu öffnen.
- 10. Ändern Sie die Eigenschaften im YAML-Code entsprechend Ihren Anforderungen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie im [„AWS Lambda Aufruf“-Aktionsreferenz](#).
- 11. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu validieren.
- 12. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

Von der Aktion „AWS LambdaAufruf“ erzeugte Variablen

Wenn die AWS Lambda Aufrufaktion ausgeführt wird, werden Variablen erzeugt, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Weitere Informationen finden Sie unter [AWS LambdaAktionsvariablen „aufrufen“](#) im [Liste der vordefinierten Variablen](#).

"AWS Lambdainvoke"-Aktionsdefinition

Die AWS Lambda Aufrufaktion ist als Satz von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie unter [„AWS Lambda Aufruf“-Aktionsreferenz](#) im [Referenz zur Workflow-Definition](#).

Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen

In diesem Abschnitt wird beschrieben, wie Sie die Aufgabendefinitionsaktion „Amazon ECS rendern“ zu Ihrem Workflow hinzufügen können. Diese Aktion aktualisiert das Image-Feld in einer Amazon Elastic Container Service (Amazon ECS) [-Aufgabendefinitionsdatei](#) mit einem Docker-Image-Namen, der von Ihrem Workflow zur Laufzeit bereitgestellt wird.



**Note**

Sie können diese Aktion auch verwenden, um das `environment` Feld der Aufgabendefinition mit Umgebungsvariablen zu aktualisieren.

## Themen

- [Wann sollte diese Aktion verwendet werden](#)
- [Funktionsweise](#)
- [Beispiel für einen Arbeitsablauf](#)
- [Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen](#)
- [Die aktualisierte Aufgabendefinitionsdatei anzeigen](#)
- [Variablen, die durch die Aktion „Amazon ECS-Aufgabendefinition rendern“ erzeugt wurden](#)
- [Aktionsdefinition „Amazon ECS-Aufgabendefinition rendern“](#)

## Wann sollte diese Aktion verwendet werden

Verwenden Sie diese Option, wenn Sie über einen Workflow verfügen, der ein Docker-Image mit dynamischen Inhalten wie einer Commit-ID oder einem Zeitstempel erstellt und markiert.

Verwenden Sie diese Aktion nicht, wenn Ihre Aufgabendefinitionsdatei einen Bildwert enthält, der immer gleich bleibt. In diesem Fall können Sie den Namen Ihres Bilds manuell in die Aufgabendefinitionsdatei eingeben.

## Funktionsweise

Sie müssen die Amazon ECS-Aufgabendefinitionsaktion rendern zusammen mit den Aktionen Build und Deploy to Amazon ECS in Ihrem Workflow verwenden. Zusammen funktionieren diese Aktionen wie folgt:

1. Die Build-Aktion erstellt Ihr Docker-Image und kennzeichnet es mit einem Namen, einer Commit-ID, einem Zeitstempel oder einem anderen dynamischen Inhalt. Ihre Build-Aktion könnte beispielsweise so aussehen:

```
MyECSWorkflow
  Actions:
    BuildAction:
```

```

Identifizier: aws/build@v1
...
Configuration:
  Steps:
    # Build, tag, and push the Docker image...
    - Run: docker build -t MyDockerImage:${WorkflowSource.CommitId} .
    ...

```

Im vorherigen Code gibt die `docker build -t` Direktive an, das Docker-Image zu erstellen und es zur Laufzeit der Aktion mit der Commit-ID zu kennzeichnen. Der generierte Image-Name könnte so aussehen:

`MyDockerImage:a37bd7e`

- Die Aktion „Amazon ECS-Aufgabendefinition rendern“ fügt Ihrer Aufgabendefinitionsdatei den dynamisch generierten Imagenamen wie folgt hinzu: `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
        }
      ]
    }
  ],
  ...
}

```

Optional können Sie auch festlegen, dass die Aktion „Amazon ECS-Aufgabendefinition rendern“ der Aufgabendefinition Umgebungsvariablen wie folgt hinzufügt:

```
{
```

```
"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"
      }
    ]
  }
],
...
}
```

Weitere Informationen zu Umgebungsvariablen finden Sie unter [Spezifizieren von Umgebungsvariablen](#) im Amazon Elastic Container Service Developer Guide.

3. Die Aktion In Amazon ECS bereitstellen registriert die aktualisierte Aufgabendefinitionsdatei bei Amazon ECS. Durch die Registrierung der aktualisierten Aufgabendefinitionsdatei wird das neue Image `MyDockerImage:a37bd7e` in Amazon ECS bereitgestellt.

### Beispiel für einen Arbeitsablauf

Im Folgenden finden Sie ein Beispiel für einen vollständigen Workflow, der die Aufgabendefinitionsaktion Amazon ECS rendern sowie die Aktionen Build und Deploy umfasst. Der Zweck des Workflows besteht darin, ein Docker-Image zu erstellen und in Ihrem Amazon ECS-Cluster bereitzustellen. Der Workflow besteht aus den folgenden Bausteinen, die nacheinander ausgeführt werden:

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine Build-Aktion (`BuildDocker`) — Beim Auslösen erstellt die Aktion das Docker-Image mithilfe der Dockerfile, kennzeichnet es mit einer Commit-ID und überträgt das Image an Amazon ECR. Weitere Informationen zur Build-Aktion finden Sie unter [Erstellen mithilfe von Workflows in CodeCatalyst](#)

- Eine Amazon ECS-Aufgabendefinitionsaktion rendern (RenderTaskDef) — Nach Abschluss der Build-Aktion aktualisiert diese Aktion ein `taskdef.json` vorhandenes Objekt im Stammverzeichnis Ihres Quell-Repositorys mit einem `image` Feldwert, der die richtige Commit-ID enthält. Sie speichert die aktualisierte Datei unter einem neuen Dateinamen (`task-definition-random-string.json`) und erstellt dann ein Ausgabeartefakt, das diese Datei enthält. Die Renderaktion generiert außerdem eine Variable namens `task-definition` und setzt sie auf den Namen der neuen Aufgabendefinitionsdatei. Das Artefakt und die Variable werden für die Aktion Deploy verwendet, die als Nächstes folgt.
- Eine Aktion „In Amazon ECS bereitstellen“ (DeployToECS) — Nach Abschluss der Aktion „Amazon ECS-Aufgabendefinition rendern“ sucht die Aktion „In Amazon ECS bereitstellen“ nach dem von der Renderaktion generierten Ausgabeartefakt (TaskDefArtifact), findet die darin enthaltene `task-definition-random-string.json` Datei und registriert sie bei Ihrem Amazon ECS-Service. Der Amazon ECS-Service folgt dann den Anweisungen in der `task-definition-random-string.json` Datei, um Amazon ECS-Aufgaben — und zugehörige Docker-Image-Container — in Ihrem Amazon ECS-Cluster auszuführen.

```
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:
```

```
Identifizier: 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}
```

## Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen

Verwenden Sie die folgenden Anweisungen, um Ihrem Workflow die Aktion „Amazon ECS rendern“ - Aufgabendefinition hinzuzufügen.

### Voraussetzung

Bevor Sie beginnen, stellen Sie sicher, dass Sie über einen Workflow verfügen, der eine Build-Aktion enthält, die dynamisch ein Docker-Image generiert. Einzelheiten finden Sie im vorherigen [Beispiel-Workflow](#).

### Visual

Um die Aktion „Amazon ECS-Aufgabendefinition rendern“ mit dem visuellen Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.

7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der Aktion „Amazon ECS-Aufgabendefinition rendern“ und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Or

- Wählen Sie „Amazon ECS-Aufgabendefinition rendern“. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
  - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „Amazon-ECS-Aufgabendefinition rendern“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um die Aktion „Amazon ECS-Aufgabendefinition rendern“ mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.

7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste Amazon aus CodeCatalyst.
9. Suchen Sie nach der Aktion „Amazon ECS-Aufgabendefinition rendern“ und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Or

- Wählen Sie „Amazon ECS-Aufgabendefinition rendern“. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
  - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „Amazon-ECS-Aufgabendefinition rendern“](#).
11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Nächste Schritte

Nachdem Sie die Render-Aktion hinzugefügt haben, fügen Sie Ihrem Workflow die Aktion Deploy to Amazon ECS hinzu, indem Sie den Anweisungen unter folgen [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#). Gehen Sie beim Hinzufügen der Bereitstellungsaktion wie folgt vor:

1. Wählen Sie auf der Registerkarte „Eingaben“ der Aktion „Bereitstellen“ unter Artefakte — optional das Artefakt aus, das durch die Renderaktion generiert wurde. Sie enthält die aktualisierte Aufgabendefinitionsdatei.

Weitere Informationen zu Artefakten finden Sie unter [Mit Artefakten arbeiten](#).

2. Geben Sie auf der Registerkarte Konfiguration der Bereitstellungsaktion im Feld Aufgabendefinition die folgende Aktionsvariable an: `${action-name.task-definition}` wobei *Aktionsname* der Name Ihrer Renderaktion ist, z. B. RenderTaskDef Die Renderaktion setzt diese Variable auf den neuen Namen der Aufgabendefinitionsdatei.



Weitere Hinweise zu Variablen finden Sie unter [Arbeiten mit Variablen](#).

Weitere Informationen zur Konfiguration der Bereitstellungsaktion finden Sie im vorherigen [Beispiel-Workflow](#).

## Die aktualisierte Aufgabendefinitionsdatei anzeigen

Sie können den Namen und den Inhalt der aktualisierten Aufgabendefinitionsdatei anzeigen.

Um den Namen der aktualisierten Aufgabendefinitionsdatei anzuzeigen, nachdem sie von der Aktion Amazon ECS-Aufgabendefinition rendern verarbeitet wurde.

1. Suchen Sie den Lauf, der eine abgeschlossene Renderaktion enthält:
  - a. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  - b. Wählen Sie Ihr Projekt.
  - c. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  - d. Wählen Sie den Namen des Workflows, der die Renderaktion enthält. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  - e. Wählen Sie einen Lauf aus, der die abgeschlossene Renderaktion enthält.
2. Wählen Sie im Workflow-Diagramm die Renderaktion aus.
3. Wählen Sie Ausgaben aus.
4. Wählen Sie Variablen.
5. Der Name der Aufgabendefinitionsdatei wird angezeigt. Er sieht ähnlich aus wie `task-definition--259-0a2r7gx1TF5X-.json`.

## Um den Inhalt der aktualisierten Aufgabendefinitionsdatei anzuzeigen

1. Suchen Sie den Lauf, der eine abgeschlossene Renderaktion enthält:
  - a. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  - b. Wählen Sie Ihr Projekt.
  - c. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.

- d. Wählen Sie den Namen des Workflows, der die Renderaktion enthält. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  - e. Wählen Sie einen Lauf aus, der die abgeschlossene Renderaktion enthält.
2. Wählen Sie in der Workflow-Ausführung oben neben Visual und YAML die Option Workflow-Ausgaben aus.
  3. Wählen Sie im Abschnitt Artefakte neben dem Artefakt, das die aktualisierte Aufgabendefinitionsdatei enthält, die Option Herunterladen aus. Für dieses Artefakt wird die Spalte Produziert von auf den Namen Ihrer Renderaktion gesetzt.
  4. Öffnen Sie die ZIP-Datei, um die JSON-Datei mit der Aufgabendefinition anzuzeigen.

Variablen, die durch die Aktion „Amazon ECS-Aufgabendefinition rendern“ erzeugt wurden

Wenn die Aktion „Amazon ECS-Aufgabendefinition rendern“ ausgeführt wird, erzeugt sie Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [Aktionsvariablen „Amazon ECS-Aufgabendefinition rendern“](#) in der [Liste der vordefinierten Variablen](#).

Aktionsdefinition „Amazon ECS-Aufgabendefinition rendern“

Die Amazon ECS-Aufgabendefinitionsaktion rendern ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „Amazon-ECS-Aufgabendefinition rendern“](#) in der [Referenz zur Workflow-Definition](#).

### GitHub Aktionen hinzufügen

Eine GitHub Aktion ist einer [CodeCatalyst Aktion](#) sehr ähnlich, außer dass sie für die Verwendung mit GitHub Workflows entwickelt wurde. Einzelheiten zu GitHub Aktionen finden Sie in der Dokumentation zu [GitHub Aktionen](#).

Sie können eine GitHub Aktion zusammen mit systemeigenen CodeCatalyst Aktionen in einem CodeCatalyst Workflow verwenden.

Es gibt zwei Möglichkeiten, einem CodeCatalyst Workflow eine GitHub Aktion hinzuzufügen:

- Sie können die GitHub Aktion aus einer kuratierten Liste in der CodeCatalyst Konsole auswählen. Es sind mehrere beliebte GitHub Aktionen verfügbar. Weitere Informationen finden Sie unter [Eine kuratierte Aktion GitHub hinzufügen](#).
- Wenn die GitHub Aktion, die Sie verwenden möchten, in der CodeCatalyst Konsole nicht verfügbar ist, können Sie sie mithilfe der Aktion GitHub Aktionen hinzufügen.

Eine GitHub Aktionen-Aktion ist eine CodeCatalyst Aktion, die eine GitHub Aktion umschließt und sie mit CodeCatalyst Workflows kompatibel macht.

Hier ist ein Beispiel für eine GitHub Actions-Aktion, die die [Super-Linter-Aktion umschließt](#): 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
```

Im vorherigen Code umschließt die Aktion CodeCatalyst GitHub Aktionen (identifiziert durch `aws/github-actions-runner@v1`) die Super-Linter-Aktion (identifiziert durch `github/super-linter@v4`), sodass sie in einem Workflow funktioniert. CodeCatalyst

Weitere Informationen finden Sie unter [Aktion „GitHub Aktionen“ hinzufügen](#).

Alle GitHub Aktionen — sowohl kuratierte als auch nicht — müssen in eine GitHub Aktionsaktion () `aws/github-actions-runner@v1` eingeschlossen werden, wie im vorherigen Beispiel gezeigt. Der Wrapper ist erforderlich, damit die Aktion ordnungsgemäß funktioniert.

## Themen

- [Wie unterscheiden sich GitHub Aktionen von CodeCatalyst Aktionen?](#)
- [Können GitHub Aktionen mit anderen CodeCatalyst Aktionen im Workflow interagieren?](#)
- [Welche GitHub Aktionen kann ich verwenden?](#)
- [Einschränkungen von GitHub Aktionen in CodeCatalyst](#)
- [Wie füge ich eine GitHub Aktion hinzu \(allgemeine Schritte\)?](#)
- [Wird die GitHub Aktion ausgeführt GitHub?](#)
- [Kann ich auch GitHub Workflows verwenden?](#)
- [Tutorial: Lint-Code mit einer GitHub Aktion](#)
- [Aktion „GitHub Aktionen“ hinzufügen](#)

- [Eine kuratierte Aktion GitHub hinzufügen](#)
- [Mit GitHub Aktions-Ausgabeparametern arbeiten](#)

Wie unterscheiden sich GitHub Aktionen von CodeCatalyst Aktionen?

GitHub Aktionen, die innerhalb eines CodeCatalyst Workflows verwendet werden, haben nicht die gleiche Zugriffs- und Integrationsebene mit AWS den CodeCatalyst Funktionen (wie [Umgebungen](#) und [Probleme](#)) wie CodeCatalyst Aktionen.

Können GitHub Aktionen mit anderen CodeCatalyst Aktionen im Workflow interagieren?

Ja. GitHub Aktionen können beispielsweise Variablen, die von anderen CodeCatalyst Aktionen erzeugt wurden, als Eingabe verwenden und auch Ausgabeparameter und Artefakte gemeinsam mit CodeCatalyst Aktionen verwenden. Weitere Informationen finden Sie unter [Mit GitHub Aktions-Ausgabeparametern arbeiten](#).

Welche GitHub Aktionen kann ich verwenden?

Sie können jede GitHub Aktion verwenden, die über die CodeCatalyst Konsole verfügbar ist, und jede GitHub Aktion, die im [GitHubMarketplace](#) verfügbar ist. Wenn Sie sich für eine GitHub Aktion aus dem Marketplace entscheiden, beachten Sie die folgenden [Einschränkungen](#).

Einschränkungen von GitHub Aktionen in CodeCatalyst

- GitHub Aktionen können nicht mit dem CodeCatalyst [Lambda-Compute-Typ](#) verwendet werden.
- GitHub Aktionen werden auf dem Docker-Image der Laufzeitumgebung [vom November 2022](#) ausgeführt, das ältere Tools enthält. Weitere Informationen über das Image und die Tools finden Sie unter [Arbeiten mit Docker-Images für die Laufzeitumgebung](#)
- GitHub Aktionen, die intern auf den [githubKontext](#) angewiesen sind oder auf GitHub spezifische Ressourcen verweisen, funktionieren nicht in CodeCatalyst. Die folgenden Aktionen funktionieren beispielsweise nicht in CodeCatalyst:
  - Aktionen, die versuchen, GitHub Ressourcen hinzuzufügen, zu ändern oder zu aktualisieren. Beispiele hierfür sind Aktionen, die Pull-Requests aktualisieren oder Probleme in verursachen GitHub.
  - Fast alle Aktionen sind unter <https://github.com/actions> aufgeführt.

- GitHub Aktionen, bei denen es sich um [Docker-Container-Aktionen](#) handelt, funktionieren, müssen jedoch vom Docker-Standardbenutzer (root) ausgeführt werden. Führen Sie die Aktion nicht als Benutzer 1001 aus. (Zum Zeitpunkt des Schreibens arbeitet der Benutzer 1001 in GitHub, aber nicht in CodeCatalyst.) Weitere Informationen finden Sie im Thema [USER](#) unter [Dockerfile-Unterstützung für GitHub Aktionen](#).

Eine Liste der über die CodeCatalyst Konsole verfügbaren GitHub Aktionen finden Sie unter [Eine kuratierte Aktion GitHub hinzufügen](#)

Wie füge ich eine GitHub Aktion hinzu (allgemeine Schritte)?

Die allgemeinen Schritte zum Hinzufügen einer GitHub Aktion zu einem CodeCatalyst Workflow lauten wie folgt:

1. In Ihrem CodeCatalyst Projekt erstellen Sie einen Workflow. In diesem Workflow definieren Sie, wie Ihre Anwendung erstellt, getestet und bereitgestellt werden soll. Weitere Informationen finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#).
2. Im Workflow fügen Sie eine kuratierte GitHub Aktion hinzu oder Sie fügen die Aktion GitHub Aktionen hinzu.
3. Sie führen einen der folgenden Schritte aus:
  - Wenn Sie eine kuratierte Aktion hinzufügen möchten, konfigurieren Sie sie. Weitere Informationen finden Sie unter [Eine kuratierte Aktion GitHub hinzufügen](#).
  - Wenn Sie eine nicht kuratierte Aktion hinzufügen möchten, fügen Sie innerhalb der Aktion GitHubAktionen den YAML-Code der GitHub Aktion ein. Du findest diesen Code auf der Detailseite deiner ausgewählten GitHub Aktion im [GitHubMarketplace](#). Sie müssen den Code wahrscheinlich leicht modifizieren, damit er funktioniert CodeCatalyst. Weitere Informationen finden Sie unter [Aktion „GitHub Aktionen“ hinzufügen](#).
4. (Optional) Innerhalb des Workflows fügen Sie weitere Aktionen wie die Build- und Testaktionen hinzu. Weitere Informationen finden Sie unter [Erstellen, Testen und Bereitstellen mit Workflows in CodeCatalyst](#).
5. Sie starten den Workflow entweder manuell oder automatisch über einen Trigger. Der Workflow führt die GitHub Aktion und alle anderen Aktionen im Workflow aus. Weitere Informationen finden Sie unter [Einen Workflow-Lauf starten](#).

Eine ausführliche Anleitung finden Sie unter:

- [Eine kuratierte Aktion GitHub hinzufügen.](#)
- [Aktion „GitHub Aktionen“ hinzufügen.](#)

Wird die GitHub Aktion ausgeführt GitHub?

Nein. Die GitHub Aktion läuft unter CodeCatalyst Verwendung CodeCatalyst der [Build-Maschinen](#).

Kann ich auch GitHub Workflows verwenden?

Nein.

Tutorial: Lint-Code mit einer GitHub Aktion

In diesem Tutorial fügen Sie die [GitHub Super-Linter-Aktion](#) zu einem CodeCatalyst Amazon-Workflow hinzu. Die Super-Linter-Aktion untersucht den Code, findet Bereiche, in denen der Code Fehler, Formatierungsprobleme und verdächtige Konstrukte aufweist, und gibt die Ergebnisse dann an die Konsole aus). CodeCatalyst Nachdem Sie den Linter zu Ihrem Workflow hinzugefügt haben, führen Sie den Workflow aus, um eine Node.js -Beispielanwendung () zu linten. `app.js` Anschließend beheben Sie die gemeldeten Probleme und führen den Workflow erneut aus, um zu überprüfen, ob die Korrekturen erfolgreich waren.

 Tip

[Erwägen Sie, Super-Linter zu verwenden, um YAML-Dateien wie Vorlagen zu verlinken. AWS CloudFormation](#)

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie ein Quell-Repository](#)
- [Schritt 2: Fügen Sie eine Datei `app.js` hinzu](#)
- [Schritt 3: Erstellen Sie einen Workflow, der die Super-Linter-Aktion ausführt](#)
- [Schritt 4: Probleme beheben, die der Super-Linter gefunden hat](#)
- [Bereinigen](#)

Voraussetzungen

Bevor Sie beginnen, benötigen Sie:

- Ein CodeCatalyst Leerzeichen mit einem verbundenen AWS-Konto. Weitere Informationen finden Sie unter [Einen Bereich erstellen, der AWS Builder ID-Benutzer unterstützt](#).
- Ein leeres Projekt in Ihrem CodeCatalyst Bereich namens `codecatalyst-linter-project`. Wählen Sie die Option Von vorne beginnen, um dieses Projekt zu erstellen.

Weitere Informationen finden Sie unter [Ein leeres Projekt in Amazon erstellen CodeCatalyst](#).

### Schritt 1: Erstellen Sie ein Quell-Repository

In diesem Schritt erstellen Sie ein Quell-Repository in CodeCatalyst. In diesem Repository speichern Sie die Quelldatei der Beispielanwendung für dieses Tutorial. `app.js`

Weitere Informationen zu Quell-Repositorys finden Sie unter [Erstellen eines Quell-Repositorys](#).

Um ein Quell-Repository zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem Projekt, `codecatalyst-linter-project`.
3. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.
4. Wählen Sie Repository hinzufügen und anschließend Repository erstellen aus.
5. Geben Sie im Feld Repository-Name Folgendes ein:

`codecatalyst-linter-source-repository`

6. Wählen Sie Erstellen aus.

### Schritt 2: Fügen Sie eine Datei `app.js` hinzu

In diesem Schritt fügen Sie Ihrem Quell-Repository eine `app.js` Datei hinzu. Der `app.js` enthält Funktionscode, der einige Fehler enthält, die der Linter finden wird.

Um die Datei `app.js` hinzuzufügen

1. Wählen Sie in der CodeCatalyst Konsole Ihr Projekt aus, `codecatalyst-linter-project`.
2. Wählen Sie im Navigationsbereich Code und dann Source Repositories aus.

3. Wählen Sie aus der Liste der Quell-Repositorys Ihr Repository aus. `codecatalyst-linter-source-repository`
4. Wählen Sie unter Dateien die Option Datei erstellen aus.
5. Geben Sie in das Textfeld den folgenden Code ein:

```
// 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. Geben Sie als Dateiname `einapp.js`. Behalten Sie die anderen Standardoptionen bei.



## 7. Wählen Sie Commit (Übergeben).

Sie haben jetzt eine Datei mit dem Namen `erstelltapp.js`.

### Schritt 3: Erstellen Sie einen Workflow, der die Super-Linter-Aktion ausführt

In diesem Schritt erstellen Sie einen Workflow, der die Super-Linter-Aktion ausführt, wenn Sie Code in Ihr Quell-Repository übertragen. Der Workflow besteht aus den folgenden Bausteinen, die Sie in einer YAML-Datei definieren:

- Ein Trigger — Dieser Trigger startet die Workflow-Ausführung automatisch, wenn Sie eine Änderung an Ihr Quell-Repository übertragen. Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).
- Eine Aktion „GitHub Aktionen“ — Beim Trigger führt die Aktion GitHub Aktionen die Super-Linter-Aktion aus, die wiederum alle Dateien in Ihrem Quell-Repository überprüft. Findet der Linter ein Problem, schlägt die Workflow-Aktion fehl.

Um einen Workflow zu erstellen, der die Super-Linter-Aktion ausführt

1. Wählen Sie in der CodeCatalyst Konsole Ihr Projekt aus, `codecatalyst-linter-project`
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie Workflow erstellen aus.
4. Wählen Sie für Quell-Repository die Option `codecatalyst-linter-source-repository`.
5. Wählen Sie für Branch die Option `main`.
6. Wählen Sie Erstellen aus.
7. Löschen Sie den YAML-Beispielcode.
8. Fügen Sie das folgende YAML hinzu:

```
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
```

Ersetzen Sie den Code im vorherigen Code *github-action-code* durch den Super-Linter-Aktionscode, wie in den folgenden Schritten dieses Verfahrens beschrieben.

9. Gehen Sie auf die [Super-Linter-Seite](#) im GitHub Marketplace.
10. Suchen Sie unter `steps :` (Kleinbuchstaben) nach dem Code und fügen Sie ihn in den CodeCatalyst Workflow unter `Steps :` (Großbuchstaben) ein.

Passen Sie den GitHub Aktionscode an die CodeCatalyst Standards an, wie im folgenden Code gezeigt.

Ihr CodeCatalyst Workflow sieht jetzt wie folgt aus:

```
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. (Optional) Wählen Sie „Validieren“, um sicherzustellen, dass der YAML-Code gültig ist, bevor Sie ihn festschreiben.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, wählen Sie Ihr `codecatalyst-linter-source-repository` Repository aus und wählen Sie erneut Commit.

Sie haben jetzt einen Workflow erstellt. Eine Workflow-Ausführung wird aufgrund des oben im Workflow definierten Triggers automatisch gestartet.

## Um die laufende Workflow-Ausführung anzuzeigen

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
2. Wählen Sie den Workflow aus, den Sie gerade erstellt haben: `codecatalyst-linter-workflow`
3. Wählen Sie im Workflow-Diagramm SuperLinterAction.
4. Warten Sie, bis die Aktion fehlschlägt. Dieser Fehler wird erwartet, weil der Linter Probleme im Code gefunden hat.
5. Lassen Sie die CodeCatalyst Konsole geöffnet und gehen Sie zu [Schritt 4: Probleme beheben, die der Super-Linter gefunden hat](#).

### Schritt 4: Probleme beheben, die der Super-Linter gefunden hat

Der Super-Linter sollte Probleme im `app.js` Code sowie in der `README.md` Datei gefunden haben, die in Ihrem Quell-Repository enthalten ist.

### Um die Probleme zu beheben, die der Linter gefunden hat

1. Wählen Sie in der CodeCatalyst Konsole die Registerkarte Logs und dann Lint Code Base aus.

Die Protokolle, die die Super-Linter-Aktion generiert hat, werden angezeigt.

2. Scrollen Sie in den Super-Linter-Protokollen nach unten bis etwa Zeile 90, wo Sie den Beginn der Probleme finden. Sie sehen etwa wie folgt aus:

```
/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. Korrigieren Sie `app.js` und `README.md` in Ihrem Quell-Repository und übernehmen Sie Ihre Änderungen.

#### Tip

Um das zu beheben `README.md`, füge `markdown` es dem Codeblock wie folgt hinzu:

```
```markdown
Setup examples:
...
```
```

Ihre Änderungen starten automatisch eine weitere Workflow-Ausführung. Warten Sie, bis der Workflow abgeschlossen ist. Wenn Sie alle Probleme behoben haben, sollte der Workflow erfolgreich sein.

## Bereinigen

Bereinigen Sie in CodeCatalyst, um Spuren dieses Tutorials aus Ihrer Umgebung zu entfernen.

Zum Aufräumen in CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Löschen `codecatalyst-linter-source-repository`.
3. Löschen `codecatalyst-linter-workflow`.

In diesem Tutorial haben Sie gelernt, wie Sie die GitHub Super-Linter-Aktion zu einem CodeCatalyst Workflow hinzufügen, um Code zu vereinfachen.

## Aktion „GitHub Aktionen“ hinzufügen

Eine GitHub Aktionsaktion ist eine CodeCatalyst Aktion, die eine GitHub Aktion umschließt und sie mit CodeCatalyst Workflows kompatibel macht.

Weitere Informationen finden Sie unter [GitHub Aktionen hinzufügen](#).

Gehen Sie folgendermaßen vor, um die Aktion GitHub Aktionen zu einem Workflow hinzuzufügen.

### Tip

Ein Tutorial, das Ihnen zeigt, wie Sie die Aktion GitHub Aktionen verwenden, finden Sie unter [Tutorial: Lint-Code mit einer GitHub Aktion](#).

## Visual

So fügen Sie die Aktion "GitHub Aktionen" mit dem visuellen Editor hinzu

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Edit (Bearbeiten) aus.
  6. Wählen Sie Visual.
  7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste die Option GitHub.
  9. Suchen Sie nach der Aktion GitHub Aktionen und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Or
- Wählen Sie GitHub Aktionen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
  10. Füllen Sie auf den Registerkarten Eingaben und Konfiguration die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „GitHub Aktionen“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), wie es sowohl im YAML- als auch im visuellen Editor angezeigt wird.
  11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
  12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um die Aktion "GitHub Aktionen" mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  2. Wählen Sie Ihr Projekt.
  3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  5. Wählen Sie Edit (Bearbeiten) aus.
  6. Wählen Sie YAML.
  7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
  8. Wählen Sie in der Drop-down-Liste die Option GitHub.
  9. Suchen Sie nach der Aktion GitHub Aktionen und führen Sie einen der folgenden Schritte aus:
    - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- Or
- Wählen Sie GitHub Aktionen aus. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld:
    - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den Quellcode der Aktion anzuzeigen.
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
  10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen verfügbaren Eigenschaften finden Sie in der [Referenz zur Aktion „GitHub Aktionen“](#).
  11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
  12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Aktionsdefinition „GitHub Aktionen“

Die Aktion „GitHub Aktionen“ ist als eine Reihe von YAML-Eigenschaften in Ihrer Workflow-Definitionsdatei definiert. Informationen zu diesen Eigenschaften finden Sie [Referenz zur Aktion „GitHub Aktionen“](#) in der [Referenz zur Workflow-Definition](#).

### Eine kuratierte Aktion GitHub hinzufügen

Eine kuratierte GitHub Aktion ist eine GitHub Aktion, die in der CodeCatalyst Konsole verfügbar gemacht wird und als Beispiel für die Verwendung einer GitHub Aktion in einem CodeCatalyst Workflow dient.

Kuratierte GitHub Aktionen sind in der [Aktion „ CodeCatalyst-authored GitHub Actions“](#) zusammengefasst, die durch die ID identifiziert wird. `aws/github-actions-runner@v1` [Die kuratierte GitHub Aktion OSS sieht zum Beispiel wie folgt aus: 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.
```

Im vorherigen Code umschließt die Aktion CodeCatalyst GitHub Aktionen (identifiziert durch `aws/github-actions-runner@v1`) die TruffleHog OSS-Aktion (identifiziert durch `truffelsecurity/trufflehog@v3.16.0`), sodass sie in einem CodeCatalyst Workflow funktioniert.

Um diese Aktion zu konfigurieren, würden Sie die leeren Zeichenfolgen unter `with:` durch Ihre eigenen Werte ersetzen. Beispiele:


**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.

Gehen Sie wie folgt vor, um einem Workflow eine kuratierte GitHub Aktion hinzuzufügen. Allgemeine Informationen zur Verwendung von GitHub Aktionen in einem CodeCatalyst Workflow finden Sie unter [GitHub Aktionen hinzufügen](#).

 **Note**

Wenn Sie Ihre GitHub Aktion nicht in der Liste der kuratierten Aktionen sehen, können Sie sie trotzdem mit der Aktion [GitHub Aktionen](#) zu Ihrem Workflow hinzufügen. Weitere Informationen finden Sie unter [Aktion „GitHub Aktionen“ hinzufügen](#).

## Visual

Um eine kuratierte GitHub Aktion mit dem visuellen Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.



5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie oben links + Aktionen, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste die Option GitHub.
9. Suchen oder suchen Sie nach einer GitHub Aktion und führen Sie einen der folgenden Schritte aus:
  - Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Or

- Wählen Sie den Namen der GitHub Aktion. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld gehen Sie wie folgt vor:
    - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
    - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
10. Füllen Sie auf den Registerkarten Eingaben, Konfiguration und Ausgaben die Felder nach Ihren Bedürfnissen aus. Eine Beschreibung der einzelnen Felder finden Sie unter [Referenz zur Aktion „GitHub Aktionen“](#). Diese Referenz enthält detaillierte Informationen zu jedem Feld (und dem entsprechenden YAML-Eigenschaftswert), das für die Aktion GitHubAktionen verfügbar ist, da es sowohl im YAML- als auch im visuellen Editor angezeigt wird.

Informationen zu den Konfigurationsoptionen, die für die kuratierte GitHub Aktion verfügbar sind, finden Sie in der zugehörigen Dokumentation.

11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um eine kuratierte GitHub Aktion mit dem YAML-Editor hinzuzufügen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wählen Sie Ihr Projekt.

3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
8. Wählen Sie in der Drop-down-Liste die Option GitHub.
9. Suchen oder suchen Sie nach einer GitHub Aktion und führen Sie einen der folgenden Schritte aus:

- Wählen Sie das Pluszeichen (+), um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.

Or

- Wählen Sie den Namen der GitHub Aktion. Das Dialogfeld mit den Aktionsdetails wird angezeigt. In diesem Dialogfeld gehen Sie wie folgt vor:
  - (Optional) Wählen Sie „Quelltext [anzeigen](#)“, um den [Quellcode der Aktion](#) anzuzeigen.
  - Wählen Sie Zum Workflow hinzufügen, um die Aktion zum Workflow-Diagramm hinzuzufügen und den zugehörigen Konfigurationsbereich zu öffnen.
- 10. Ändern Sie die Eigenschaften im YAML-Code nach Ihren Bedürfnissen. Eine Erläuterung der einzelnen Eigenschaften, die für die Aktion GitHub Aktionen verfügbar sind, finden Sie in [Referenz zur Aktion „GitHub Aktionen“](#)

Informationen zu den Konfigurationsoptionen, die für die kuratierte GitHub Aktion verfügbar sind, finden Sie in der zugehörigen Dokumentation.

11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

Mit GitHub Aktions-Ausgabeparametern arbeiten

Sie können [GitHub Ausgabeparameter](#) in Ihre CodeCatalyst Workflows integrieren.

**Note**

Ein anderes Wort für Ausgabeparameter ist variabel. Da in der Dokumentation der Begriff Ausgabeparameter GitHub verwendet wird, werden wir diesen Begriff auch verwenden.

## Themen

- [Exportieren eines GitHub Ausgabeparameters, damit er von anderen Aktionen verwendet werden kann](#)
- [Referenzieren eines GitHub Ausgabeparameters](#)

Exportieren eines GitHub Ausgabeparameters, damit er von anderen Aktionen verwendet werden kann

Verwenden Sie die folgenden Anweisungen, um einen GitHub Ausgabeparameter zu exportieren, sodass er für die Verwendung in anderen CodeCatalyst Workflow-Aktionen verfügbar ist.

Um einen GitHub Ausgabeparameter zu exportieren

1. Öffnen Sie einen Workflow und wählen Sie Bearbeiten. Detaillierte Anweisungen finden Sie unter [Um einen Workflow zu bearbeiten](#).
2. Fügen Sie in der Aktion GitHub Aktionen, die den Ausgabeparameter generiert, den Sie exportieren möchten, einen Outputs Abschnitt mit einer zugrunde liegenden Variables Eigenschaft hinzu, die wie folgt aussieht:

```
Actions:
  MyGitHubAction:
    Identifier: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'step-id_output-name'
```

Ersetze:

- *Schritt-ID* mit dem Wert der `id`: Eigenschaft im Abschnitt der GitHub steps Aktion.
- *Ausgabename mit dem Namen des Ausgabeparameters*. GitHub

## Beispiel

Das folgende Beispiel zeigt Ihnen, wie Sie einen GitHub Ausgabeparameter namens exportieren. 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
```

## Referenzieren eines GitHub Ausgabeparameters

Verwenden Sie die folgenden Anweisungen, um auf einen GitHub Ausgabeparameter zu verweisen.

Um auf einen GitHub Ausgabeparameter zu verweisen

1. Führen Sie die Schritte unter [Exportieren eines GitHub Ausgabeparameters, damit er von anderen Aktionen verwendet werden kann](#) aus.

Der GitHub Ausgabeparameter ist jetzt für die Verwendung in anderen Aktionen verfügbar.

2. Notieren Sie sich den Variables Wert des Ausgabeparameters. Er enthält einen Unterstrich (\_).
3. Verwenden Sie für den Ausgabeparameter die folgende Syntax:

```
${action-name.output-name}
```

Ersetze:

- *Aktionsname* mit dem Namen der CodeCatalyst GitHub Aktion, die den Ausgabeparameter erzeugt (verwenden Sie nicht das name oder id der GitHub Aktion).

- **Ausgabename** mit dem Variables Wert des Ausgabeparameters, den Sie zuvor notiert haben.

## Beispiel

```
BuildActionB:
  Identifier: aws/build@v1
  Configuration:
    Steps:
      - Run: echo `${MyGitHubAction.random-color-generator_SELECTEDCOLOR}
```

## Beispiel mit Kontext

Das folgende Beispiel zeigt Ihnen, wie Sie eine SELECTEDCOLOR Variable eingebenGitHubActionA, ausgeben und dann in darauf verweisenBuildActionB.

```
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}
```

## Eine Aktion entfernen

Verwenden Sie die folgenden Anweisungen, um eine Aktion aus einem Workflow zu entfernen.

## Visual

Um eine Aktion mit dem Visual Editor zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm in der Aktion, die Sie entfernen möchten, das vertikale Ellipsensymbol und dann Entfernen aus.
8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um eine Aktion mit dem YAML-Editor zu entfernen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Suchen Sie den Abschnitt der YAML, der die Aktion enthält, die Sie entfernen möchten.

Wählen Sie den Abschnitt aus und drücken Sie die Löschtaste auf Ihrer Tastatur.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Entwicklung einer benutzerdefinierten Aktion

Mit dem Action Development Kit (ADK) können Sie eine benutzerdefinierte CodeCatalyst Aktion entwickeln, die Sie in Ihren Workflows verwenden können. Anschließend können Sie die Aktion im CodeCatalyst Aktionskatalog veröffentlichen, sodass andere CodeCatalyst Benutzer sie anzeigen und in ihren Workflows verwenden können.

Um eine Aktion zu entwickeln, zu testen und zu veröffentlichen (Aufgaben auf hoher Ebene)

1. Installieren Sie die erforderlichen Tools und Pakete, die für die Entwicklung einer Aktion erforderlich sind.
2. Erstellen Sie ein CodeCatalyst Repository zum Speichern Ihres Aktionscodes.
3. Initialisieren Sie die Aktion. Dadurch werden die für die Aktion erforderlichen Quelldateien festgelegt, einschließlich einer Aktionsdefinitionsdatei (`action.yml`), die Sie mit Ihrem eigenen Code aktualisieren können.
4. Führen Sie ein Bootstrapping für den Aktionscode durch, um die Tools und Bibliotheken zu erhalten, die zum Erstellen, Testen und Veröffentlichen des Aktionsprojekts erforderlich sind.
5. Erstellen Sie die Aktion auf Ihrem lokalen Computer und übertragen Sie die Änderungen in Ihr CodeCatalyst Repository.
6. Testen Sie die Aktion lokal mit Komponententests und führen Sie den vom ADK generierten Workflow in aus. CodeCatalyst
7. Veröffentlichen Sie die Aktion im CodeCatalyst Aktionskatalog, indem Sie in der Konsole auf die Schaltfläche Veröffentlichen klicken. CodeCatalyst

Eine ausführliche Anleitung finden Sie im [Amazon CodeCatalyst Action Development Kit Developer Guide](#).

## Gruppierung von Aktionen in Aktionsgruppen

Eine Aktionsgruppe enthält eine oder mehrere Aktionen. Das Gruppieren von Aktionen in Aktionsgruppen hilft Ihnen dabei, Ihren Arbeitsablauf zu organisieren, und ermöglicht es Ihnen auch, Abhängigkeiten zwischen verschiedenen Aktionsgruppen zu konfigurieren.

### Note

Aktionsgruppen können nicht innerhalb anderer Aktionen oder Aktionsgruppen verschachtelt werden.

Verwenden Sie die folgenden Anweisungen, um eine Aktionsgruppe zu definieren.

### Visual

Nicht verfügbar. Wählen Sie YAML, um die YAML-Anweisungen anzuzeigen.

### YAML

Um eine Aktionsgruppe zu definieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in Code hinzuActions, der dem folgenden ähnelt:

```
Actions:
  action-group-name:
    Actions:
      action-1:
        Identifier: aws/build@v1
        Configuration:
```



```
    ...
  action-2:
    Identifier: aws/ecs-deploy@v1
    Configuration:
    ...
```

Ein weiteres Beispiel finden Sie unter [Beispiel: Definition von zwei Aktionsgruppen](#). Weitere Informationen finden Sie in der Beschreibung der `action-group-name` Immobilie in [Aktionen](#) der [Referenz zur Workflow-Definition](#).

- (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
- Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

### Beispiel: Definition von zwei Aktionsgruppen

Das folgende Beispiel zeigt, wie zwei Aktionsgruppen definiert werden: `BuildAndTest` und `Deploy`. Die `BuildAndTest` Aktionsgruppe umfasst zwei Aktionen (`Build` und `Test`), und die `Deploy` Aktionsgruppe umfasst auch zwei Aktionen (`DeployCloudFormationStack` und `DeployToECS`).

```
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:
        ...
```

## Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen

Wenn Sie einem Workflow Aktionen hinzufügen, werden diese standardmäßig im [Visual Editor](#) nebeneinander hinzugefügt. Das bedeutet, dass die Aktionen parallel ausgeführt werden, wenn Sie einen Workflow-Lauf starten. Wenn Sie möchten, dass Aktionen in sequentieller Reihenfolge ausgeführt werden (und vertikal im Visual Editor angezeigt werden), müssen Sie Abhängigkeiten zwischen ihnen einrichten. Sie können beispielsweise eine Test Aktion so einrichten, dass sie von der Build Aktion abhängt, sodass die Testaktion nach der Build-Aktion ausgeführt wird.

Sie können Abhängigkeiten zwischen Aktionen, zwischen Aktionsgruppen sowie zwischen Aktionen und Aktionsgruppen einrichten. Sie können one-to-many Abhängigkeiten auch so konfigurieren, dass eine Aktion von mehreren anderen abhängt, bevor sie gestartet wird. Konsultieren Sie die [Richtlinien für die Einrichtung von Abhängigkeiten](#), um sicherzustellen, dass Ihr Abhängigkeits-Setup der YAML-Syntax des Workflows entspricht.

### Themen

- [Abhängigkeiten einrichten](#)
- [Richtlinien für die Einrichtung von Abhängigkeiten](#)
- [Beispiele](#)

### Abhängigkeiten einrichten

Verwenden Sie die folgenden Anweisungen, um Abhängigkeiten zwischen Aktionen in einem Workflow einzurichten.

### Visual

So richten Sie Abhängigkeiten mit dem Visual Editor ein

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.

7. Wählen Sie im Workflow-Diagramm die Aktion aus, die von einer anderen Aktion abhängen soll.
8. Wählen Sie die Registerkarte Eingaben.
9. Gehen Sie unter Abhängig von — optional wie folgt vor:

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Hinweise zur Funktion „Hängt davon ab“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#)

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um Abhängigkeiten mit dem YAML-Editor einzurichten

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer Aktion, die von einer anderen Aktion abhängt, Code hinzu, der dem folgenden ähnelt:

```
action-name:
  DependsOn:
    - action-1
```

Weitere Beispiele finden Sie unter [Beispiele](#). Allgemeine Richtlinien finden Sie unter [Richtlinien für die Einrichtung von Abhängigkeiten](#). Weitere Informationen finden Sie in

der Beschreibung der DependsOn Immobilie im Abschnitt [Referenz zur Workflow-Definition](#) für Ihre Aktion.

- (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
- Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Richtlinien für die Einrichtung von Abhängigkeiten

Beachten Sie bei der Konfiguration von Abhängigkeiten die folgenden Richtlinien:

- Wenn sich eine Aktion innerhalb einer Aktionsgruppe befindet, kann diese Aktion nur von anderen Aktionen innerhalb derselben Aktionsgruppe abhängen.
- Aktionen und Aktionsgruppen können von anderen Aktionen und Aktionsgruppen auf derselben Ebene in der YAML-Hierarchie abhängen, jedoch nicht auf einer anderen Ebene.

## Beispiele

Die folgenden Beispiele zeigen, wie Abhängigkeiten zwischen Aktionen und Aktionsgruppen in der Workflow-Definitionsdatei konfiguriert werden.

## Themen

- [Beispiel: Konfiguration einer einfachen Abhängigkeit](#)
- [Beispiel: Konfiguration einer Aktionsgruppe so, dass sie von einer Aktion abhängt](#)
- [Beispiel: Konfiguration einer Aktionsgruppe so, dass sie von einer anderen Aktionsgruppe abhängt](#)
- [Beispiel: Konfiguration einer Aktionsgruppe so, dass sie von mehreren Aktionen abhängt](#)

## Beispiel: Konfiguration einer einfachen Abhängigkeit

Das folgende Beispiel zeigt, wie eine Test Aktion so konfiguriert wird, dass sie von der Build Aktion abhängt, die die DependsOn Eigenschaft verwendet.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
```

```

...
Test:
  DependsOn:
  - Build
  Identifier: aws/managed-test@v1
  Configuration:
  ...

```

Beispiel: Konfiguration einer Aktionsgruppe so, dass sie von einer Aktion abhängt

Das folgende Beispiel zeigt, wie eine `DeployGroup` Aktionsgruppe so konfiguriert wird, dass sie von der `FirstAction` Aktion abhängt. Beachten Sie, dass sich Aktion und Aktionsgruppe auf derselben Ebene befinden.

```

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:
      ...

```

Beispiel: Konfiguration einer Aktionsgruppe so, dass sie von einer anderen Aktionsgruppe abhängt

Das folgende Beispiel zeigt, wie eine `DeployGroup` Aktionsgruppe so konfiguriert wird, dass sie von der `BuildAndTestGroup` Aktionsgruppe abhängt. Beachten Sie, dass sich die Aktionsgruppen auf derselben Ebene befinden.

```

Actions:
  BuildAndTestGroup: # Action group 1
    Actions:
      BuildAction:
      ...
      TestAction:
      ...
  DeployGroup: #Action group 2

```

**DependsOn:**

- **BuildAndTestGroup**

## Actions:

DeployAction1:

...

DeployAction2:

...

Beispiel: Konfiguration einer Aktionsgruppe so, dass sie von mehreren Aktionen abhängt

Das folgende Beispiel zeigt, wie eine DeployGroup Aktionsgruppe so konfiguriert wird, dass sie von der SecondAction Aktion, der Aktion und der BuildAndTestGroup Aktionsgruppe abhängt. FirstAction Beachten Sie, dass DeployGroup sich das auf derselben Ebene wie FirstActionSecondAction, und befindetBuildAndTestGroup.

## 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:

...


## Den Quellcode einer Aktion anzeigen

Sie können den Quellcode einer Aktion einsehen, um sicherzustellen, dass er keinen riskanten Code, Sicherheitslücken oder andere Fehler enthält.


Folgen Sie den folgenden Anweisungen, um den Quellcode einer Aktion [CodeCatalyst](#), einer Aktion von [CodeCatalyst Labs](#) oder eines [Drittanbieters einzusehen](#).

 Note

Um den Quellcode einer [GitHubAktion](#) einzusehen, gehen Sie auf die Seite der Aktion im [GitHub Marketplace](#). Die Seite enthält einen Link zum Repository der Aktion, wo Sie den Quellcode der Aktion finden können.

 Note

Sie können den Quellcode der folgenden CodeCatalyst Aktionen nicht anzeigen: [Build](#), [Test](#), [GitHub Actions](#).

 Note

AWSunterstützt oder garantiert nicht den Aktionscode von GitHub Aktionen oder Aktionen Dritter.

Um den Quellcode einer Aktion einzusehen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Suchen Sie die Aktion, deren Code Sie sich ansehen möchten:
  - a. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  - b. Wählen Sie den Namen eines beliebigen Workflows oder erstellen Sie einen. Informationen zum Erstellen eines Workflows finden Sie unter [Einen Workflow erstellen, bearbeiten und löschen](#).
  - c. Wählen Sie Edit (Bearbeiten) aus.
  - d. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.
  - e. Wählen Sie in der Drop-down-Liste Amazon CodeCatalyst zum Ansehen CodeCatalyst, CodeCatalyst Labs und Aktionen von Drittanbietern aus.

- f. Suchen Sie nach einer Aktion und wählen Sie ihren Namen. Wählen Sie nicht das Pluszeichen (+).

Details zur Aktion werden angezeigt.

4. Wählen Sie im Dialogfeld mit den Aktionsdetails unten die Option Herunterladen aus.

Es wird eine Seite mit dem Amazon S3 S3-Bucket angezeigt, in dem sich der Quellcode der Aktion befindet. Informationen zu Amazon S3 finden Sie unter [Was ist Amazon S3?](#) im Amazon Simple Storage Service-Benutzerhandbuch.

5. Prüfen Sie den Code, um sicherzustellen, dass er Ihren Qualitäts- und Sicherheitserwartungen entspricht.

## Mit Action-Versionen arbeiten

Wenn Sie einem Workflow eine Aktion hinzufügen, CodeCatalyst fügt Amazon der Workflow-Definitionsdatei standardmäßig die Vollversion im folgenden Format hinzu:

*vmajor.minor.patch*

Beispiele:

```
My-Build-Action:  
  Identifier: aws/build@v1.0.0
```

Sie können die Vollversion in der `Identifier` Eigenschaft kürzen, sodass der Workflow immer die neueste Neben- oder Patch-Version der Aktion verwendet.

Wenn Sie beispielsweise Folgendes angeben:

```
My-CloudFormation-Action:  
  Identifier: aws/cfn-deploy@v1.0
```

... und die neueste Patch-Version ist `1.0.4`, dann wird die Aktion verwendet `1.0.4`. Wenn zum Beispiel eine neuere Version veröffentlicht wird `1.0.5`, dann wird die Aktion verwendet `1.0.5`. Wenn beispielsweise eine Nebenversion veröffentlicht wird `1.1.0`, wird die Aktion weiterhin verwendet `1.0.5`.

Ausführliche Anweisungen zur Angabe von Versionen finden Sie in einem der folgenden Themen.

Themen



- [Angabe der zu verwendenden Aktionsversion](#)
- [Ermitteln, welche Versionen einer Aktion verfügbar sind](#)

## Angabe der zu verwendenden Aktionsversion

Verwenden Sie die folgenden Anweisungen, um anzugeben, welche Version einer Aktion Ihr Workflow verwenden soll. Sie können die neueste Haupt- oder Nebenversion oder eine bestimmte Patch-Version angeben.

Wir empfehlen, die neueste Neben- oder Patch-Version einer Aktion zu verwenden.

## Visual


Nicht verfügbar. Wählen Sie YAML, um die YAML-Anweisungen anzuzeigen.

## YAML


Um einen Workflow so zu konfigurieren, dass er die neueste Version einer Aktion oder eine bestimmte Patch-Version verwendet

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Suchen Sie die Aktion, deren Version Sie bearbeiten möchten.
8. Suchen Sie die `Identifier` Eigenschaft der Aktion und legen Sie die Version auf eine der folgenden Optionen fest:
  - `action-identifizier @v major` — Verwenden Sie diese Syntax, damit der Workflow eine bestimmte Hauptversion verwendet und die neuesten Neben- und Patch-Versionen automatisch ausgewählt werden können.
  - `action-identifizier @v major.minor` — Verwenden Sie diese Syntax, damit der Workflow eine bestimmte Nebenversion verwendet und die neueste Patch-Version automatisch ausgewählt wird.

- *action-identifizier @v major. unbedeutend. Patch* — Verwenden Sie diese Syntax, damit der Workflow eine bestimmte Patch-Version verwendet.

 Note

Wenn Sie nicht sicher sind, welche Versionen verfügbar sind, finden Sie weitere Informationen unter [Ermitteln, welche Versionen einer Aktion verfügbar sind](#).

 Note

Sie können die Hauptversion nicht weglassen.

9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Ermitteln, welche Versionen einer Aktion verfügbar sind

Ermitteln Sie anhand der folgenden Anweisungen, welche Versionen einer Aktion für Sie in einem Workflow verfügbar sind.

### Visual

Um festzustellen, welche Aktionsversionen verfügbar sind

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Suchen Sie die Aktion, deren Versionen Sie sich ansehen möchten:
  - a. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  - b. Wählen Sie den Namen eines beliebigen Workflows oder erstellen Sie einen. Informationen zum Erstellen eines Workflows finden Sie unter [Einen Workflow erstellen, bearbeiten und löschen](#).
  - c. Wählen Sie Edit (Bearbeiten) aus.
  - d. Wählen Sie links oben + Aktionen aus, um den Aktionskatalog zu öffnen.

- e. Wählen Sie in der Drop-down-Liste Amazon CodeCatalyst zum Ansehen CodeCatalyst, CodeCatalyst Labs und Aktionen von Drittanbietern oder wählen Sie, ob Sie kuratierte GitHub Aktionen anzeigen GitHubmöchten.
- f. Suchen Sie nach einer Aktion und wählen Sie ihren Namen. Wählen Sie nicht das Pluszeichen (+).

Details zur Aktion werden angezeigt.

4. Wählen Sie im Dialogfeld mit den Aktionsdetails oben rechts die Dropdownliste Versionen aus, um eine Liste der verfügbaren Versionen der Aktion anzuzeigen.

## YAML

Nicht verfügbar. Wählen Sie „visuell“, um die Anweisungen des visuellen Editors anzuzeigen.

## Mit Artefakten arbeiten

Ein Artefakt ist die Ausgabe einer Workflow-Aktion und besteht in der Regel aus einem Ordner oder Archiv mit Dateien. Artefakte sind wichtig, weil sie es Ihnen ermöglichen, Dateien und Informationen zwischen Aktionen gemeinsam zu nutzen.

Möglicherweise haben Sie eine Build-Aktion, die eine `sam-template.yml` Datei generiert, Sie möchten jedoch, dass eine Bereitstellungsaktion diese verwendet. In diesem Szenario würden Sie ein Artefakt verwenden, um es der Build-Aktion zu ermöglichen, die `sam-template.yml` Datei mit der Bereitstellungsaktion gemeinsam zu nutzen. Der Code könnte etwa so aussehen:

```
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
```


Im vorherigen Code generiert die Build-Aktion (BuildAction) eine `sam-template.yml` Datei und fügt sie dann einem Ausgabeartefakt namens MYARTIFACT hinzu. Eine nachfolgende Bereitstellungsaktion (DeployAction) wird MYARTIFACT als Eingabe angegeben, sodass sie auf die `sam-template.yml` Datei zugreifen kann.

## Themen

- [Kann ich Artefakte teilen, ohne sie als Ausgaben und Eingaben anzugeben?](#)
- [Kann ich Artefakte zwischen Workflows teilen?](#)
- [Definition eines Ausgabeartefakts](#)
- [Definieren Sie ein Eingabeartefakt](#)
- [Referenzieren von Dateien in einem Artefakt](#)
- [Artefakte werden heruntergeladen](#)
- [Beispiele](#)

## Kann ich Artefakte teilen, ohne sie als Ausgaben und Eingaben anzugeben?

Ja, Sie können Artefakte zwischen Aktionen gemeinsam nutzen, ohne sie in den Inputs Abschnitten Outputs und im YAML-Code Ihrer Aktionen anzugeben. Dazu müssen Sie Compute Sharing aktivieren. Weitere Informationen zu Compute Sharing und zur Angabe von Artefakten, wenn es aktiviert ist, finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

 Note

Mit der Compute Sharing-Funktion können Sie zwar den YAML-Code Ihres Workflows vereinfachen, indem Sie die Inputs Abschnitte Outputs und und überflüssig machen, es gibt jedoch Einschränkungen, die Sie beachten sollten, bevor Sie sie aktivieren. Informationen zu diesen Einschränkungen finden Sie unter [Überlegungen zur gemeinsamen Nutzung von Rechenleistung](#).

## Kann ich Artefakte zwischen Workflows teilen?

Nein, Sie können Artefakte nicht zwischen verschiedenen Workflows gemeinsam nutzen. Sie können Artefakte jedoch zwischen Aktionen innerhalb desselben Workflows gemeinsam nutzen.

## Definition eines Ausgabeartefakts

Verwenden Sie die folgenden Anweisungen, um ein Artefakt zu definieren, das von einer Aktion ausgegeben werden soll. Dieses Artefakt steht dann anderen Aktionen zur Verfügung.

### Note

Nicht alle Aktionen unterstützen Ausgabeartefakte. Um festzustellen, ob Ihre Aktion sie unterstützt, führen Sie die folgenden Anweisungen für den visuellen Editor durch und prüfen Sie, ob die Aktion auf der Registerkarte Ausgaben die Schaltfläche Ausgabeartefakte enthält. Falls ja, werden Ausgabeartefakte unterstützt.

## Visual

Um ein Ausgabeartefakt mit dem visuellen Editor zu definieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, die das Artefakt erzeugen soll.
8. Wählen Sie die Registerkarte Outputs.
9. Wählen Sie unter Artefakte die Option Artefakt hinzufügen aus.
10. Wählen Sie Artefakt hinzufügen und geben Sie wie folgt Informationen in die Felder ein.

Name des Build-Artefakts

Geben Sie den Namen eines Artefakts an, das durch die Aktion generiert wurde. Artefaktnamen müssen innerhalb eines Workflows eindeutig sein und sind auf alphanumerische Zeichen (a-z, A-Z, 0-9) und Unterstriche (\_) beschränkt. Leerzeichen, Bindestriche (-) und andere Sonderzeichen sind nicht zulässig. Sie können keine Anführungszeichen verwenden, um Leerzeichen, Bindestriche und andere Sonderzeichen in Namen von Ausgabeartefakten zuzulassen.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#)

### Von build erstellte Dateien

Geben Sie die Dateien an, die in dem Artefakt CodeCatalyst enthalten sind, das durch die Aktion ausgegeben wird. Diese Dateien werden durch die Workflow-Aktion generiert, wenn sie ausgeführt wird, und sind auch in Ihrem Quell-Repository verfügbar. Dateipfade können sich in einem Quell-Repository oder einem Artefakt aus einer früheren Aktion befinden und sind relativ zum Quell-Repository oder Artefakt-Stamm. Sie können Glob-Muster verwenden, um Pfade anzugeben. Beispiele:


- Um eine einzelne Datei anzugeben, die sich im Stammverzeichnis Ihres Build-Speicherorts oder Quell-Repository-Speicherorts befindet, verwenden `my-file.jar`.
- Um eine einzelne Datei in einem Unterverzeichnis anzugeben, verwenden Sie `directory/my-file.jar` oder `directory/subdirectory/my-file.jar`.
- Um alle Dateien anzugeben, verwenden Sie `**/*`. Das `**` Glob-Muster gibt an, dass es einer beliebigen Anzahl von Unterverzeichnissen entspricht.
- Um alle Dateien und Verzeichnisse in einem Verzeichnis mit dem Namen `directory` anzugeben, verwenden Sie `directory/**/*`. Das `**` Glob-Muster gibt an, dass es einer beliebigen Anzahl von Unterverzeichnissen entspricht.
- Um alle Dateien in einem Verzeichnis mit dem Namen `directory`, aber nicht in einem seiner Unterverzeichnisse anzugeben, verwenden Sie `directory/*`

#### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder ein anderes Sonderzeichen enthält, schließen Sie den Pfad in doppelte Anführungszeichen (") ein. "" Weitere

Hinweise zu Sonderzeichen finden Sie unter [Richtlinien und Konventionen zur Syntax](#)

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

 Note

Möglicherweise müssen Sie dem Dateipfad ein Präfix hinzufügen, um anzugeben, in welchem Artefakt oder in welcher Quelle das Objekt gefunden werden soll. Weitere Informationen finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um ein Ausgabeartefakt mit dem YAML-Editor zu definieren

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer Workflow-Aktion Code hinzu, der dem folgenden ähnelt:

```
action-name:  
  Outputs:  
    Artifacts:  
      - Name: artifact-name
```

Files:

- *file-path-1*
- *file-path-2*

Weitere Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für Ihre Aktion.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Definieren Sie ein Eingabeartefakt

Wenn Sie ein Artefakt verwenden möchten, das durch eine andere Aktion generiert wurde, müssen Sie es als Eingabe für die aktuelle Aktion angeben. Möglicherweise können Sie mehrere Artefakte als Eingabe angeben — das hängt von der Aktion ab. Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für Ihre Aktion.

### Note

Sie können nicht auf Artefakte aus anderen Workflows verweisen.

Verwenden Sie die folgenden Anweisungen, um ein Artefakt aus einer anderen Aktion als Eingabe für die aktuelle Aktion anzugeben.

### Voraussetzung

Bevor Sie beginnen, stellen Sie sicher, dass Sie das Artefakt aus der anderen Aktion ausgegeben haben. Weitere Informationen finden Sie unter [Definition eines Ausgabeartefakts](#). Wenn Sie das Artefakt ausgeben, steht es anderen Aktionen zur Verfügung.

### Visual

Um ein Artefakt als Eingabe für eine Aktion anzugeben (visueller Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.



4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, für die Sie ein Artefakt als Eingabe angeben möchten.
8. Wählen Sie Eingaben.
9. Gehen Sie unter Artefakte — optional wie folgt vor:

Geben Sie Artefakte aus früheren Aktionen an, die Sie als Eingabe für diese Aktion bereitstellen möchten. Diese Artefakte müssen bereits in früheren Aktionen als Ausgabeartefakte definiert sein.

Wenn Sie keine Eingabeartefakte angeben, müssen Sie mindestens ein Quell-Repository unter angeben `action-name/Inputs/Sources`.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

#### Note

Wenn die Dropdownliste Artefakte — optional nicht verfügbar ist (visueller Editor) oder wenn Sie bei der Validierung Ihres YAML (YAML-Editors) Fehler erhalten, kann das daran liegen, dass die Aktion nur eine Eingabe unterstützt. Versuchen Sie in diesem Fall, die Quelleingabe zu entfernen.

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um ein Artefakt als Eingabe für eine Aktion anzugeben (YAML-Editor)

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Fügen Sie in der Aktion, in der Sie das Artefakt als Eingabe angeben möchten, Code hinzu, der dem folgenden ähnelt:

```
action-name:  
  Inputs:  
  Artifacts:  
    - artifact-name
```

Weitere Beispiele finden Sie unter [Beispiele](#).

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Referenzieren von Dateien in einem Artefakt

Wenn Sie über eine Datei verfügen, die sich in einem Artefakt befindet, und Sie in einer Ihrer Workflow-Aktionen auf diese Datei verweisen müssen, gehen Sie wie folgt vor.

### Note

Siehe auch [Verweisen auf Dateien in einem Quell-Repository](#).

## So verweisen Sie auf Dateien in einem Artefakt

- Fügen Sie in der Aktion, in der Sie auf eine Datei verweisen möchten, Code hinzu, der dem folgenden ähnelt:

```
Actions:
```

```
My-action:
  Inputs:
    Sources:
      - WorkflowSource
    Artifacts:
      - artifact-name
  Configuration:
    Steps:
      - run: cd $CATALYST_SOURCE_DIR_artifact-name/build-output && cat file.txt
```

Im vorherigen Code sucht die Aktion im build-output Verzeichnis im *Artefaktnamen-Artefakt* nach der Datei und zeigt sie an. file.txt

Weitere Beispiele finden Sie unter [Beispiele](#).

#### Note

Je nachdem, wie Sie Ihre Aktion konfiguriert haben, können Sie das `$CATALYST_SOURCE_DIR_artifact-name/` Präfix möglicherweise weglassen. Weitere Informationen finden Sie in der folgenden Anleitung.

Hinweise zur Bezugnahme auf Variablen:

- Wenn Ihre Aktion nur ein Element unter umfasst Inputs (z. B. ein Eingabeartefakt und keine Quelle), können Sie das Präfix weglassen und nur den Dateipfad relativ zum Artefaktstamm angeben.
- Sie können das Präfix auch weglassen, wenn sich die Datei in der primären Eingabe befindet. Die primäre Eingabe ist entweder das oder das WorkflowSource erste aufgeführte Eingabeartefakt, falls es keines gibt. WorkflowSource
- Das Präfix kann je nach der Aktion, die Sie verwenden, unterschiedlich sein. Weitere Informationen können Sie der folgenden Tabelle entnehmen.

| Aktionstyp   | Zu verwendendes Dateipfad präfix  | Beispiel  |
|--|---|---|
| <a href="#">Aktion erstellen</a> , <a href="#">Aktion testen</a> | <code>\$CATALYST_SOURCE_D<br/>IR_ <i>artifact-name</i> /</code>   | <code>\$CATALYST_SOURCE_D<br/>IR_MyArtifact/folder1/file.txt</code>   |
| Alle anderen Aktionen  | <code>\$CATALYST_SOURCE_D<br/>IR_ <i>artifact-name</i> /<br/><br/>or<br/><br/>/artifacts/ <i>current-action-name</i> /<i>artifact-name</i> /<i>(Dieser Pfad ist ein symbolischer Link zu \$CATALYST_SOURCE_D IR_ artifact-name/)</i></code> | <code>\$CATALYST_SOURCE_D<br/>IR_MyArtifact/folder1/file.txt<br/><br/>or<br/><br/>/artifacts/MyCurrentAction/MyArtifact/folder1/file.txt</code> |

## Artefakte werden heruntergeladen

Sie können Artefakte, die durch Ihre Workflow-Aktionen generiert wurden, zur Fehlerbehebung herunterladen und untersuchen. Es gibt zwei Arten von Artefakten, die Sie herunterladen können:

- Quellartefakte — Ein Artefakt, das einen Snapshot des Inhalts des Quell-Repositorys enthält, so wie er zu Beginn der Ausführung vorhanden war.
- Workflow-Artefakte — Ein Artefakt, das in der `Outputs` Eigenschaft der Konfigurationsdatei Ihres Workflows definiert ist.

Um die vom Workflow ausgegebenen Artefakte herunterzuladen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.

5. Wählen Sie unter dem Namen des Workflows die Option Runs aus.
6. Wählen Sie im Ausführungsverlauf in der Spalte Lauf-ID einen Lauf aus. z. B. Run-95a4d.
7. Wählen Sie unter dem Namen des Laufs die Option Artifacts aus.
8. Wähle neben einem Artefakt die Option Herunterladen aus. Eine Archivdatei wird heruntergeladen. Ihr Dateiname besteht aus sieben zufälligen Zeichen.
9. Extrahieren Sie das Archiv mit einem Archivextraktionsprogramm Ihrer Wahl.

## Beispiele

Die folgenden Beispiele zeigen, wie Artefakte in der Workflow-Definitionsdatei ausgegeben und referenziert werden.

### Themen

- [Beispiel: Ein Artefakt ausgeben](#)
- [Beispiel: Eingabe eines Artefakts, das durch eine andere Aktion generiert wurde](#)
- [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#)
- [Beispiel: Referenzieren einer Datei in einem einzigen Artefakt](#)
- [Beispiel: Verweisen auf eine Datei in einem Artefakt, wenn a vorhanden ist WorkflowSource](#)

### Beispiel: Ein Artefakt ausgeben

Das folgende Beispiel zeigt, wie ein Artefakt ausgegeben wird, das zwei .jar-Dateien enthält.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ARTIFACT1
          Files:
            - build-output/file1.jar
            - build-output/file2.jar
```

### Beispiel: Eingabe eines Artefakts, das durch eine andere Aktion generiert wurde

Das folgende Beispiel zeigt Ihnen, wie Sie ein aufgerufenes Artefakt ausgeben und ARTIFACT4 in BuildActionA dieses eingeben. 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:
```

### Beispiel: Referenzieren von Dateien in mehreren Artefakten

Das folgende Beispiel zeigt `BuildActionC`, wie Sie zwei Artefakte mit dem Namen `ART5` und `ART6` in ausgeben und dann auf zwei Dateien mit den Namen `file5.txt` (in `ArtifactART5`) und `file6.txt` (in `ArtifactART6`) in `BuildActionD` (unter `Steps`) verweisen.

#### Note

Weitere Informationen zum Referenzieren von Dateien finden Sie unter [Referenzieren von Dateien in einem Artefakt](#)

#### Note

Das Beispiel zeigt zwar, dass das `$CATALYST_SOURCE_DIR_ART5` Präfix verwendet wird, Sie könnten es aber weglassen. Das liegt daran, dass `ART5` es die primäre Eingabe ist. Weitere Informationen zur primären Eingabe finden Sie unter [Referenzieren von Dateien in einem Artefakt](#).

```
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
```

### Beispiel: Referenzieren einer Datei in einem einzigen Artefakt

Das folgende Beispiel zeigt, wie Sie ein Artefakt mit dem Namen ART7 in BuildActionE ausgeben und dann auf file7.txt (im ArtefaktART7) in BuildActionF (unter) verweisen. Steps

Beachten Sie, dass die Referenz nicht das Präfix für den \$CATALYST\_SOURCE\_DIR\_ *Artefaktnamen* vor dem build-output Verzeichnis benötigt, wie dies in der Fall war. [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#) Dies liegt daran, dass unter nur ein Element angegeben ist. Inputs

#### Note

Weitere Informationen zum Verweisen auf Dateien finden Sie unter [Referenzieren von Dateien in einem Artefakt](#).

```
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
```

Beispiel: Verweisen auf eine Datei in einem Artefakt, wenn a vorhanden ist WorkflowSource

Das folgende Beispiel zeigt, wie Sie ein Artefakt mit dem Namen ART8 in BuildActionG ausgeben und dann auf file8.txt (in ArtefaktART8) in BuildActionH (unter) verweisen. Steps

Beachten Sie, dass für die Referenz das Präfix für den \$CATALYST\_SOURCE\_DIR\_ *Artefaktnamen* erforderlich ist, wie in. [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#) Das liegt daran, dass unter Inputs (eine Quelle und ein Artefakt) mehrere Elemente angegeben sind. Sie benötigen also das Präfix, um anzugeben, wo nach der Datei gesucht werden soll.

#### Note

Weitere Informationen zum Verweisen auf Dateien finden Sie unter. [Referenzieren von Dateien in einem Artefakt](#)

```
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
```

## Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen

In einem CodeCatalyst Workflow können Sie das Image der Rechen- und Laufzeitumgebung angeben, das zur Ausführung von Workflow-Aktionen CodeCatalyst verwendet wird.

Compute bezieht sich auf die Rechenengine (die CPU, den Arbeitsspeicher und das Betriebssystem), die CodeCatalyst zur Ausführung von Workflow-Aktionen verwaltet und gewartet wird.

Ein Laufzeitumgebungs-Image ist ein Docker-Container, in dem Workflow-Aktionen CodeCatalyst ausgeführt werden. Der Docker-Container wird auf der von Ihnen ausgewählten Rechenplattform ausgeführt und umfasst ein Betriebssystem und zusätzliche Tools, die für eine Workflow-Aktion möglicherweise erforderlich sind, z. B. Node.js und .tar. AWS CLI

### Themen

- [Mit Compute arbeiten](#)
- [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#)
- [Arbeiten mit Docker-Images für die Laufzeitumgebung](#)

### Mit Compute arbeiten

Compute bezieht sich auf die Computing-Engine (die CPU, den Arbeitsspeicher und das Betriebssystem), die CodeCatalyst zur Ausführung von Workflow-Aktionen verwaltet und gewartet wird.

#### Note

Wenn Compute als Eigenschaft des Workflows definiert ist, kann es nicht als Eigenschaft einer Aktion in diesem Workflow definiert werden. Ebenso kann Compute, wenn es als Eigenschaft einer Aktion definiert ist, nicht im Workflow definiert werden.

### Themen

- [Über Berechnungstypen](#)
- [Über Rechenflotten](#)
- [Flotteneigenschaften auf Abruf](#)
- [Bereitgestellte Flotteneigenschaften](#)
- [Eine bereitgestellte Flotte erstellen, bearbeiten und löschen](#)
- [Einer Aktion eine bereitgestellte Flotte oder eine On-Demand-Rechenleistung zuweisen](#)

## Über Berechnungstypen

CodeCatalyst bietet die folgenden Berechnungstypen:

- Amazon EC2
- AWS Lambda

Amazon EC2 bietet optimierte Flexibilität bei Aktionsläufen und Lambda bietet optimierte Startgeschwindigkeiten von Aktionen. Lambda unterstützt aufgrund einer geringeren Startlatenz schnellere Workflow-Aktionsausführungen. Mit Lambda können Sie grundlegende Workflows ausführen, mit denen serverlose Anwendungen mit gemeinsamen Laufzeiten erstellt, getestet und bereitgestellt werden können. Zu diesen Laufzeiten gehören Node.js, Python, Java, .NET und Go. Es gibt jedoch einige Anwendungsfälle, die Lambda nicht unterstützt, und wenn sie Sie betreffen, verwenden Sie den Amazon EC2 EC2-Compute-Typ:

- Lambda unterstützt keine Laufzeitumgebungsbilder aus einer bestimmten Registrierung.
- Lambda unterstützt keine Tools, für die Root-Rechte erforderlich sind. Verwenden Sie für Tools wie yum oder rpm den Datentyp Amazon EC2 oder andere Tools, für die keine Root-Rechte erforderlich sind.
- Lambda unterstützt keine Docker-Builds oder -Läufe. Die folgenden Aktionen, die Docker-Images verwenden, werden nicht unterstützt: AWS CloudFormation Stack bereitstellen, In Amazon ECS bereitstellen, Veröffentlichen in Amazon S3, AWS CDK Bootstrap, AWS CDK Bereitstellen, AWS Lambda Aufrufen und Aktionen. GitHub Docker-basierte GitHub Aktionen, die innerhalb CodeCatalyst GitHub der Actions-Aktion ausgeführt werden, werden von Lambda Compute ebenfalls nicht unterstützt. Sie können Alternativen verwenden, für die keine Root-Rechte erforderlich sind, z. B. Podman.
- Lambda unterstützt das Schreiben in Dateien außerhalb /tmp nicht. Bei der Konfiguration Ihrer Workflow-Aktionen können Sie Ihre Tools neu konfigurieren, um sie zu installieren oder in die Sie

schreiben möchten. /tmp Wenn Sie über eine Build-Aktion verfügen, die installiert wirdnpm, stellen Sie sicher, dass Sie sie für die Installation unter konfigurieren. /tmp

- Lambda unterstützt keine Laufzeiten von mehr als 15 Minuten.

## Über Rechenflotten

CodeCatalyst bietet die folgenden Rechenflotten an:

- Flotten auf Abruf
- Bereitgestellte Flotten

Bei On-Demand-Flotten stellt der Workflow beim Start einer Workflow-Aktion die benötigten Ressourcen bereit. Die Maschinen werden zerstört, wenn die Aktion abgeschlossen ist. Sie zahlen nur für die Anzahl der Minuten, für die Sie Ihre Aktionen ausführen. On-Demand-Flotten werden vollständig verwaltet und verfügen über automatische Skalierungsfunktionen zur Bewältigung von Nachfragespitzen.

CodeCatalyst bietet auch bereitgestellte Flotten an, die Maschinen enthalten, die von Amazon EC2 betrieben werden und von verwaltet werden. CodeCatalyst Bei bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen ausführen. Mit bereitgestellten Flotten sind Ihre Maschinen immer in Betrieb und es fallen Kosten an, solange sie bereitgestellt werden.

Um eine Flotte zu erstellen, zu aktualisieren oder zu löschen, benötigen Sie die Rolle eines Space-Administrators oder eines Projektadministrators.

## Flotteneigenschaften auf Abruf

CodeCatalyst bietet die folgenden On-Demand-Flotten:

| Name              | Betriebssystem | Architektur | vCPUs | Arbeitsspeicher (GiB) | Festplattenkapazität | Unterstützte Compute-Typen |
|-------------------|----------------|-------------|-------|-----------------------|----------------------|----------------------------|
| Linux.Arm64.Large | Amazon Linux 2 | Arm64       | 2     | 4                     | 64 GB                | Amazon EC2                 |

| Name                 | Betriebssystem | Architektur | vCPUs | Arbeitsspeicher (GiB) | Festplattenkapazität | Unterstützte Compute-Typen |
|----------------------|----------------|-------------|-------|-----------------------|----------------------|----------------------------|
|                      |                |             |       |                       | 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                     |
| Linux.x86-64.2XLarge | Amazon Linux 2 | x86-64      | 8     | 16                    | 128 GB               | Amazon EC2                 |

Wenn keine Flotte ausgewählt ist, CodeCatalyst verwendet `Linux.x86-64.Large`.

### Bereitgestellte Flotteneigenschaften


Eine bereitgestellte Flotte enthält die folgenden Eigenschaften:

#### Betriebssystem

Das Betriebssystem. Die folgenden Betriebssysteme sind verfügbar:

- Amazon Linux 2

- Windows Server 2022

 Note

Windows-Flotten werden nur in der Build-Aktion unterstützt. Andere Aktionen unterstützen Windows derzeit nicht.

## Architektur

Die Prozessorarchitektur. Die folgenden Architekturen sind verfügbar:

- x86\_64
- Arm64

## Typ der Maschine

Der Maschinentyp für jede Instanz. Die folgenden Maschinentypen sind verfügbar:

| vCPUs | Arbeitsspeicher (GiB) | Festplattenkapazität | Betriebssystem                        |
|-------|-----------------------|----------------------|---------------------------------------|
| 2     | 4                     | 64 GB                | Amazon Linux 2                        |
| 4     | 8                     | 128 GB               | Amazon Linux 2<br>Windows Server 2022 |
| 8     | 16                    | 128 GB               | Amazon Linux 2<br>Windows Server 2022 |

## Capacity (Kapazität)

Die anfängliche Anzahl der Maschinen, die der Flotte zugewiesen wurden. Sie definiert die Anzahl der Aktionen, die parallel ausgeführt werden können.

## Skalierungsmodus

Definiert das Verhalten, wenn die Anzahl der Aktionen die Flottenkapazität überschreitet.

## Stellen Sie bei Bedarf zusätzliche Kapazität bereit

Zusätzliche Maschinen werden bei Bedarf eingerichtet, die bei Ausführung neuer Aktionen automatisch hochskaliert und nach Abschluss der Aktionen wieder auf die Basiskapazität herunterskaliert werden. Dadurch können zusätzliche Kosten entstehen, da Sie für jeden laufenden Computer minutengenau zahlen.

## Warten Sie, bis zusätzliche Flottenkapazität verfügbar ist

Aktionsläufe werden in eine Warteschlange gestellt, bis eine Maschine verfügbar ist. Dadurch werden zusätzliche Kosten begrenzt, da keine zusätzlichen Maschinen zugewiesen werden.

## Eine bereitgestellte Flotte erstellen, bearbeiten und löschen

Folgen Sie den folgenden Anweisungen, um bereitgestellte Flotten zu erstellen, zu bearbeiten und zu löschen.

### Note

Bereitgestellte Flotten werden nach 2 Wochen Inaktivität deaktiviert. Wenn sie erneut verwendet werden, werden sie automatisch reaktiviert. Diese Reaktivierung kann jedoch zu einer Latenz führen.

## Um eine bereitgestellte Flotte zu erstellen


1. Wählen Sie im Navigationsbereich CI/CD und dann Compute aus.
2. Wählen Sie Create Provisioned Fleet aus.
3. Geben Sie im Textfeld Name der bereitgestellten Flotte einen Namen für Ihre Flotte ein.
4. Wählen Sie im Dropdownmenü Betriebssystem das Betriebssystem aus.
5. Wählen Sie im Drop-down-Menü Maschinentyp den Maschinentyp für Ihre Maschine aus.
6. Geben Sie im Textfeld Kapazität die maximale Anzahl von Maschinen in der Flotte ein.
7. Wählen Sie im Drop-down-Menü Skalierungsmodus das gewünschte Überlaufverhalten aus. Weitere Informationen zu diesen Feldern finden Sie unter [Bereitgestellte Flotteneigenschaften](#).
8. Wählen Sie Erstellen.

Nachdem Sie die bereitgestellte Flotte erstellt haben, können Sie sie einer Aktion zuweisen. Weitere Informationen finden Sie unter [Einer Aktion eine bereitgestellte Flotte oder eine On-Demand-Rechenleistung zuweisen](#).

Um eine bereitgestellte Flotte zu bearbeiten

1. Wählen Sie im Navigationsbereich CI/CD und dann Compute aus.
2. Wählen Sie in der Liste Bereitgestellte Flotte die Flotte aus, die Sie bearbeiten möchten.
3. Wählen Sie Bearbeiten aus.
4. Geben Sie im Textfeld Kapazität die maximale Anzahl von Maschinen in der Flotte ein.
5. Wählen Sie im Drop-down-Menü Skalierungsmodus das gewünschte Überlaufverhalten aus. Weitere Informationen zu diesen Feldern finden Sie unter [Bereitgestellte Flotteneigenschaften](#).
6. Wählen Sie Speichern.

Um eine bereitgestellte Flotte zu löschen

 Warning

Bevor Sie eine bereitgestellte Flotte löschen, entfernen Sie sie aus allen Aktionen, indem Sie die `Fleet` Eigenschaft aus dem YAML-Code der Aktion löschen. Jede Aktion, die nach dem Löschen weiterhin auf eine bereitgestellte Flotte verweist, schlägt fehl, wenn die Aktion das nächste Mal ausgeführt wird.

1. Wählen Sie im Navigationsbereich CI/CD und dann Compute aus.
2. Wählen Sie in der Liste Bereitgestellte Flotte die Flotte aus, die Sie löschen möchten.
3. Wählen Sie Löschen aus.
4. Geben Sie **eindelete**, um den Löschvorgang zu bestätigen.
5. Wählen Sie Löschen aus.

Einer Aktion eine bereitgestellte Flotte oder eine On-Demand-Rechenleistung zuweisen

Standardmäßig verwenden Workflow-Aktionen die `Linux.x86-64.Large` On-Demand-Flotte mit einem Amazon EC2 EC2-Berechnungstyp. Wenn Sie stattdessen eine bereitgestellte Flotte oder eine andere On-Demand-Flotte verwenden möchten `Linux.x86-64.2XLarge`, verwenden Sie beispielsweise die folgenden Anweisungen.

## Visual

Bevor Sie beginnen

- Wenn Sie eine bereitgestellte Flotte zuweisen möchten, müssen Sie zuerst die bereitgestellte Flotte erstellen. Weitere Informationen finden Sie unter [Eine bereitgestellte Flotte erstellen, bearbeiten und löschen](#).

Um einer Aktion eine bereitgestellte Flotte oder einen anderen Flottentyp zuzuweisen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, der Sie Ihre bereitgestellte Flotte oder Ihren neuen Flottentyp zuweisen möchten.
8. Wählen Sie die Registerkarte Konfiguration aus.
9. Gehen Sie in Compute Fleet wie folgt vor:

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden sollen. Bei bedarfsgesteuerten Flotten stellt der Workflow zu Beginn einer Aktion die benötigten Ressourcen bereit, und die Maschinen werden zerstört, wenn die Aktion abgeschlossen ist. Beispiele für Flotten auf Abruf: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Weitere Informationen zu Flotten auf Abruf finden Sie unter [Flotteneigenschaften auf Abruf](#)

Bei bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen, um Ihre Workflow-Aktionen auszuführen. Diese Maschinen bleiben inaktiv und können sofort Aktionen ausführen. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#)

Wenn Fleet es weggelassen wird, ist die Standardeinstellung `Linux.x86-64.Large`



10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Bevor Sie beginnen

- Wenn Sie eine bereitgestellte Flotte zuweisen möchten, müssen Sie zuerst die bereitgestellte Flotte erstellen. Weitere Informationen finden Sie unter [Eine bereitgestellte Flotte erstellen, bearbeiten und löschen](#).

Um einer Aktion eine bereitgestellte Flotte oder einen anderen Flottentyp zuzuweisen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern. /
5. Wählen Sie Bearbeiten aus.
6. Wähle YAML.
7. Suchen Sie die Aktion, der Sie Ihre bereitgestellte Flotte oder Ihren neuen Flottentyp zuweisen möchten.
8. Fügen Sie in der Aktion eine Compute Eigenschaft hinzu und legen Sie Fleet sie auf den Namen Ihrer Flotte oder Ihres On-Demand-Flottentyps fest. Weitere Informationen finden Sie in der Beschreibung der Fleet Immobilie im Abschnitt [Aktionsreferenz zum Erstellen und Testen](#) Für Ihre Aktion.
9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Rechenleistung für mehrere Aktionen gemeinsam nutzen

### Themen

- [Mehrere Aktionen auf gemeinsam genutztem Computer ausführen](#)
- [Überlegungen zur gemeinsamen Nutzung von Rechenleistung](#)
- [Compute Sharing einschalten](#)
- [Beispiele](#)

Standardmäßig werden Aktionen in einem Workflow auf separaten Instanzen in einer [Flotte](#) ausgeführt. Dieses Verhalten sorgt für isolierte und vorhersehbare Aktionen in Bezug auf den Status der Eingaben. Das Standardverhalten erfordert eine explizite Konfiguration, um Kontext wie Dateien und Variablen zwischen Aktionen gemeinsam zu nutzen.

Compute Sharing ist eine Funktion, mit der Sie alle Aktionen in einem Workflow auf derselben Instanz ausführen können. Die Nutzung von Compute Sharing kann zu schnelleren Workflow-Laufzeiten führen, da weniger Zeit für die Bereitstellung von Instanzen aufgewendet wird. Sie können Dateien (Artefakte) auch ohne zusätzliche Workflow-Konfiguration zwischen Aktionen gemeinsam nutzen.

Wenn ein Workflow mithilfe von Compute Sharing ausgeführt wird, ist eine Instanz in der Standardflotte oder der angegebenen Flotte für die Dauer aller Aktionen in diesem Workflow reserviert. Wenn der Workflow-Lauf abgeschlossen ist, wird die Instanzreservierung freigegeben.

### Mehrere Aktionen auf gemeinsam genutztem Computer ausführen

Sie können das `Compute` Attribut in der Definition YAML auf Workflow-Ebene verwenden, um sowohl die Flotten- als auch die Compute-Sharing-Eigenschaften von Aktionen anzugeben. Sie können Computing-Eigenschaften auch mit dem Visual Editor in CodeCatalyst konfigurieren. Um eine Flotte anzugeben, legen Sie den Namen einer vorhandenen Flotte fest, legen Sie den Berechnungstyp auf EC2 fest und aktivieren Sie Compute Sharing.

#### Note

Compute Sharing wird nur unterstützt, wenn der `Compute`-Typ auf EC2 festgelegt ist, und sie wird für das Betriebssystem Windows Server 2022 nicht unterstützt. Weitere Informationen zu Rechenflotten, Compute-Typen und Eigenschaften finden Sie unter [Mit Compute arbeiten](#)

**Note**

Wenn Sie das kostenlose Kontingent nutzen und die `Linux.x86-64.2XLarge` Flotte `Linux.x86-64.XLarge` oder manuell in der Workflow-Definition YAML angeben, wird die Aktion trotzdem auf der Standardflotte (`Linux.x86-64.Large`) ausgeführt. Weitere Informationen zur Verfügbarkeit von Rechenleistung und zu den Preisen finden Sie in der [Tabelle mit den Tarifoptionen](#).

Wenn Compute Sharing aktiviert ist, wird der Ordner, der die Workflow-Quelle enthält, automatisch in alle Aktionen kopiert. Sie müssen keine Ausgabeartefakte konfigurieren und sie in einer Workflow-Definition (YAML-Datei) nicht als Eingabeartefakte referenzieren. Als Workflow-Autor müssen Sie Umgebungsvariablen mithilfe von Eingaben und Ausgaben verknüpfen, genau wie Sie es ohne Compute Sharing tun würden. Wenn Sie Ordner für Aktionen außerhalb der Workflow-Quelle gemeinsam nutzen möchten, sollten Sie das Zwischenspeichern von Dateien in Betracht ziehen. Weitere Informationen erhalten Sie unter [Mit Artefakten arbeiten](#) und [Arbeiten mit Datei-Caching](#).

Das Quell-Repository, in dem sich Ihre Workflow-Definitionsdatei befindet, ist anhand der Bezeichnung gekennzeichnet. `WorkflowSource` Bei Verwendung von Compute Sharing wird die Workflow-Quelle in der ersten Aktion heruntergeladen, die darauf verweist, und automatisch für nachfolgende Aktionen im Workflow-Lauf zur Verfügung gestellt. Alle Änderungen, die durch eine Aktion an dem Ordner, der die Workflow-Quelle enthält, vorgenommen werden, z. B. durch Hinzufügen, Ändern oder Entfernen von Dateien, sind auch in den nachfolgenden Aktionen im Workflow sichtbar. Sie können in jeder Ihrer Workflow-Aktionen auf Dateien verweisen, die sich im Workflow-Quellordner befinden, genauso wie Sie es ohne Compute Sharing tun können. Weitere Informationen finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#).

**Note**

Compute-Sharing-Workflows müssen eine strikte Reihenfolge von Aktionen angeben, sodass parallel Aktionen nicht festgelegt werden können. Ausgabeartefakte können zwar bei jeder Aktion in der Sequenz konfiguriert werden, Eingabeartefakte werden jedoch nicht unterstützt.

## Überlegungen zur gemeinsamen Nutzung von Rechenleistung

Sie können Workflows mit Compute Sharing ausführen, um Workflow-Ausführungen zu beschleunigen und den Kontext zwischen Aktionen in einem Workflow, die dieselbe Instanz

verwenden, gemeinsam zu nutzen. Beachten Sie Folgendes, um festzustellen, ob die Nutzung von Compute Sharing für Ihr Szenario geeignet ist:

|   | Compute Sharing   | Ohne gemeinsame Nutzung von Rechenleistung  |
|---|---|---|
| Datenverarbeitung                               | Amazon EC2  | Amazon EC2, AWS Lambda  |
| Bereitstellung von Instanzen                    | Aktionen werden auf derselben Instanz ausgeführt  | Aktionen werden auf separaten Instanzen ausgeführt  |
| Betriebssystem                                  | Amazon Linux 2  | Amazon Linux 2, Windows Server 2022 (nur Build-Aktion)  |
| Dateien referenzieren                           | <code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code>                | <code>\$CATALYST_SOURCE_DIR_WorkflowSource , /sources/WorkflowSource/</code>                    |
| Workflow-Struktur                               | Aktionen können nur sequentiell ausgeführt werden   | Aktionen können parallel ausgeführt werden  |
| Zugriff auf Daten über Workflow-Aktionen hinweg | Greifen Sie auf die zwischengespeicherte Workflow-Quelle zu ( <code>WorkflowSource</code> ) | Greifen Sie auf Ausgaben gemeinsam genutzter Artefakte zu (erfordert zusätzliche Konfiguration) |

## Compute Sharing einschalten

Gehen Sie wie folgt vor, um Compute Sharing für einen Workflow zu aktivieren.

### Visual

Um Compute Sharing mit dem Visual Editor zu aktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.

4. Wählen Sie den Namen Ihres Workflows.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie Workflow-Eigenschaften.
8. Wählen Sie im Dropdownmenü Berechnungstyp die Option EC2 aus.
9. (Optional) Wählen Sie im Dropdownmenü Compute fleet — optional eine Flotte aus, die Sie für die Ausführung von Workflow-Aktionen verwenden möchten. Sie können eine On-Demand-Flotte wählen oder eine bereitgestellte Flotte erstellen und auswählen. Weitere Informationen finden Sie unter [Eine bereitgestellte Flotte erstellen, bearbeiten und löschen](#) und [Einer Aktion eine bereitgestellte Flotte oder eine On-Demand-Rechenleistung zuweisen](#)
10. Schalten Sie den Schalter um, um Compute Sharing zu aktivieren und die Aktionen im Workflow auf derselben Flotte ausführen zu lassen.
11. (Optional) Wählen Sie den Ausführungsmodus für den Workflow. Weitere Informationen finden Sie unter [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#).
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um Compute Sharing mit dem YAML-Editor zu aktivieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Aktivieren Sie Compute Sharing und setzen Sie das SharedInstance Feld auf TRUE und Type bis EC2. Stellen Fleet Sie eine Rechenflotte ein, die Sie zum Ausführen von Workflow-Aktionen verwenden möchten. Sie können eine On-Demand-Flotte wählen oder eine bereitgestellte Flotte erstellen und auswählen. Weitere Informationen finden Sie unter [Eine bereitgestellte Flotte erstellen, bearbeiten und löschen](#) und [Einer Aktion eine bereitgestellte Flotte oder eine On-Demand-Rechenleistung zuweisen](#)

Fügen Sie in einem Workflow-YAML Code hinzu, der dem folgenden ähnelt:

```
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. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Festschreiben zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Beispiele

### Themen

- [Beispiel: Amazon S3 Publish](#)

### Beispiel: Amazon S3 Publish

Die folgenden Workflow-Beispiele zeigen, wie Sie die Amazon S3 S3-Aktion „Veröffentlichen“ auf zwei Arten ausführen können: zuerst mithilfe von Eingabeartefakten und dann mithilfe von Compute Sharing. Bei Compute Sharing werden die Eingabeartefakte nicht benötigt, da Sie auf die zwischengespeicherten `WorkflowSource` zugreifen können. Außerdem wird das Ausgabeartefakt in der Build-Aktion nicht mehr benötigt. Die S3-Aktion „Veröffentlichen“ ist so konfiguriert, dass sie die explizite `DependsOn` Eigenschaft zur Verwaltung sequentieller Aktionen verwendet. Die Build-Aktion muss erfolgreich ausgeführt werden, damit die S3-Aktion „Veröffentlichen“ ausgeführt werden kann.

- Ohne Compute-Sharing müssen Sie Eingabeartefakte verwenden und die Ausgaben mit nachfolgenden Aktionen teilen:

```
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: dev-bucket
```

- Wenn Sie Compute Sharing mit der Einstellung `SharedInstance` auf `verwendenTRUE`, können Sie mehrere Aktionen auf derselben Instanz ausführen und Artefakte gemeinsam nutzen, indem Sie eine einzige Workflow-Quelle angeben. Eingabeartefakte sind nicht erforderlich und können nicht angegeben werden:

```
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: dev-bucket
```

## Arbeiten mit Docker-Images für die Laufzeitumgebung

Ein Laufzeitumgebungs-Image ist ein Docker-Container, in dem Workflow-Aktionen CodeCatalyst ausgeführt werden. Der Docker-Container wird auf der von Ihnen ausgewählten Rechenplattform ausgeführt und umfasst ein Betriebssystem und zusätzliche Tools, die für eine Workflow-Aktion möglicherweise erforderlich sind, z. B. Node.js und .tar. AWS CLI

Standardmäßig werden Workflow-Aktionen auf einem der [aktiven Images](#) ausgeführt, die von bereitgestellt und verwaltet werden. CodeCatalyst Nur Build- und Testaktionen unterstützen benutzerdefinierte Images. Weitere Informationen finden Sie unter [Zuweisen eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung zu einer Aktion](#).

### Themen

- [Aktive Bilder](#)



- [Was ist, wenn ein aktives Bild nicht die Tools enthält, die ich benötige?](#)
- [Zuweisen eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung zu einer Aktion](#)
- [Beispiele](#)

## Aktive Bilder

Aktive Images sind Runtime-Umgebungs-Images, die vollständig von Tools unterstützt werden CodeCatalyst und vorinstallierte Tools enthalten. Derzeit gibt es zwei Gruppen aktiver Images: eines wurde im März 2024 veröffentlicht, das andere wurde im November 2022 veröffentlicht.

Ob für eine Aktion ein Bild vom März 2024 oder vom November 2022 verwendet wird, hängt von der Aktion ab:

- Build- und Testaktionen, die am oder nach dem 26. März 2024 zu einem Workflow hinzugefügt werden, enthalten in ihrer YAML-Definition einen `Container` Abschnitt, der explizit ein [Bild vom März 2024](#) spezifiziert. Sie können den `Container` Abschnitt optional entfernen, um zum Bild vom [November 2022](#) zurückzukehren.
- Build- und Testaktionen, die vor dem 26. März 2024 zu einem Workflow hinzugefügt wurden, enthalten keinen `Container` Abschnitt in ihrer YAML-Definition und verwenden daher ein Image [vom November 2022](#). Sie können das Image vom November 2022 behalten oder es aktualisieren. Um das Image zu aktualisieren, öffnen Sie die Aktion im Visual Editor, wählen Sie die Registerkarte Konfiguration und wählen Sie dann das Bild vom März 2024 aus der Dropdownliste Docker-Image der Laufzeitumgebung aus. Durch diese Auswahl wird der YAML-Definition der Aktion ein `Container` Abschnitt hinzugefügt, der mit dem entsprechenden Bild vom März 2024 gefüllt ist.
- Für alle anderen Aktionen wird ein [Bild vom November 2022](#) verwendet, unabhängig davon, wann sie dem Workflow hinzugefügt wurden. Ein Upgrade dieser Aktionen zur Verwendung eines Images vom März 2024 ist derzeit nicht möglich.

## Themen

- [Bilder vom März 2024](#)
- [Bilder vom November 2022](#)

## Bilder vom März 2024

Die Bilder vom März 2024 sind die neuesten Bilder, die von bereitgestellt wurden CodeCatalyst. Pro Kombination aus Rechnertyp und Flotte gibt es ein Bild vom März 2024.

Die folgende Tabelle zeigt die Tools, die auf jedem Image vom März 2024 installiert sind.

### Image-Tools vom März 2024

| Tool            | CodeCatalyst<br>Amazon EC2 für<br>Linux x86_64<br>- CodeCatalystLinux_<br>x86_64:2024_03 | CodeCatalyst<br>Lambda für<br>Linux x86_64<br>- CodeCatalystLinuxL<br>ambda_x86<br>_64:2024_03 | CodeCatalyst<br>Amazon EC2<br>für Linux Arm64<br>- CodeCatalystLinux_<br>Arm64:2024_03 | CodeCatalyst<br>Lambda für<br>Arm64 - CodeCatalystLinux_<br>ambda_Ar<br>64:2024_03 |
|-----------------|--|--|--|--|
| AWS CLI         | 2.15.17  | 2,15,17  | 2,15,17  | 2,15,17  |
| AWS Copilot CLI | 1.32.1   | 1,32,1   | 1,32,1   | 1,32,1   |
| Docker          | 24,9   | N/A  | 24,9   | N/A  |
| Docker Compose  | 2.23.3   | N/A  | 2.23.3   | N/A  |
| 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            | Korretto 17  | Korretto 17  | Korretto 17  | Korretto 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   | 3.11.6   | 3.11.6   | 3.11.6   |
| pip             | 22.3.1   | 22.3.1   | 22.3.1   | 22.3.1   |
| .NET            | 8.0.100  | 8,0,100  | 8,0,100  | 8,0,100  |

## Bilder vom November 2022

Pro Kombination aus Rechnertyp und Flotte gibt es ein Bild vom November 2022. Es ist auch ein Windows-Image vom November 2022 mit der Build-Aktion verfügbar, wenn Sie eine [bereitgestellte](#) Flotte konfiguriert haben.

In der folgenden Tabelle sind die Tools aufgeführt, die auf den einzelnen Images vom November 2022 installiert sind.

### Image-Tools für November 2022

| Tool            | CodeCatalyst<br>Amazon EC2 für<br>Linux x86_64<br>- CodeCatalystLinux_<br>x86_64:2022_11 | CodeCatalyst<br>Lambda für<br>Linux x86_64<br>- CodeCatalystLinuxL<br>ambda_x86_64:2022_11 | CodeCatalyst<br>Amazon EC2<br>für Linux Arm64<br>- CodeCatalystLinux_<br>Arm64:2022_11 | CodeCatalyst<br>Lambda für<br>Arm64 - CodeCatalystLinux_<br>ambda_Ar<br>64:2022_11 |
|-----------------|--|--|--|--|
| AWS CLI         | 2.15,17  | 2,15,17  | 2,15,17  | 2,15,17  |
| AWS Copilot CLI | 0.6.0  | 0.6.0  | N/A  | N/A  |
| Docker          | 23,01  | N/A  | 23,0,1   | N/A  |
| Docker Compose  | 2.16.0   | N/A  | 2.16.0   | N/A  |
| Git             | 2.40.0   | 2.40.0   | 2,39,2   | 2,39,2   |
| Go              | 1.20.2   | 1.20.2   | 1,20,1   | 1,20,1   |
| Gradle          | 8.0.2  | 8.0.2  | 8.0.1  | 8.0.1  |
| Java            | Korretto 17  | Korretto 17  | Korretto 17  | Korretto 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   |

| Tool    | CodeCatalyst<br>Amazon EC2 für<br>Linux x86_64<br>- CodeCatalystLinux_<br>x86_64:2022_11 | CodeCatalyst<br>Lambda für<br>Linux x86_64<br>- CodeCatalystLinuxL<br>ambda_x86<br>_64:2022_11 | CodeCatalyst<br>Amazon EC2<br>für Linux Arm64<br>- CodeCatalystLinux_<br>Arm64:2022_11 | CodeCatalyst<br>Lambda für<br>Linux Arm64 - CodeCatalystLinux_<br>ambda_Ar<br>64:2022_11 |
|---------|--|--|--|--|
| Python3 | N/A  | 3.9,15   | N/A  | 3.11.2   |
| pip     | 22.2.2   | 22.2.2   | 23.0.1   | 23.0.1   |
| .NET    | 6,0,407  | 6,0,407  | 6,0,406  | 6,0,406  |

Was ist, wenn ein aktives Bild nicht die Tools enthält, die ich benötige?

Wenn keines der von Ihnen bereitgestellten [aktiven Bilder](#) die Tools CodeCatalyst enthält, die Sie benötigen, haben Sie mehrere Optionen:

- Sie können ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung bereitstellen, das die erforderlichen Tools enthält. Weitere Informationen finden Sie unter [Zuweisen eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung zu einer Aktion](#).

#### Note

Wenn Sie ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung bereitstellen möchten, stellen Sie sicher, dass in Ihrem benutzerdefinierten Image Git installiert ist.

- Sie können die Build- oder Testaktion Ihres Workflows die Tools installieren lassen, die Sie benötigen.

Sie könnten beispielsweise die folgenden Anweisungen in den Steps Abschnitt des YAML-Codes der Build- oder Testaktion aufnehmen:

Configuration:

Steps:

- Run: `./setup-script`

Die *Setup-Script-Anweisung* würde dann das folgende Skript ausführen, um den Node-Paketmanager (npm) zu installieren:

```
#!/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
```

Weitere Hinweise zur Build-Aktion YAML finden Sie unter [Aktionsreferenz zum Erstellen und Testen](#)

Zuweisen eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung zu einer Aktion

Wenn Sie kein von bereitgestelltes [Active-Image](#) verwenden möchten CodeCatalyst, können Sie ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung bereitstellen. Wenn Sie ein benutzerdefiniertes Image bereitstellen möchten, stellen Sie sicher, dass Git darin installiert ist. Das Image kann sich in Docker Hub, Amazon Elastic Container Registry oder einem beliebigen öffentlichen Repository befinden.

Informationen zum Erstellen eines benutzerdefinierten Docker-Images finden Sie unter [Containerisieren einer Anwendung](#) in der Docker-Dokumentation.

Verwenden Sie die folgenden Anweisungen, um Ihr benutzerdefiniertes Docker-Image für die Laufzeitumgebung einer Aktion zuzuweisen. Nachdem Sie ein Image angegeben haben, CodeCatalyst wird es auf Ihrer Rechenplattform bereitgestellt, wenn die Aktion gestartet wird.

#### Note

Die folgenden Aktionen unterstützen keine Docker-Images für benutzerdefinierte Laufzeitumgebungen: AWS CloudFormation Stack bereitstellen, In ECS bereitstellen und GitHub Aktionen. Docker-Images für benutzerdefinierte Laufzeitumgebungen unterstützen auch nicht den Lambda-Compute-Typ.

## Visual

So weisen Sie mit dem Visual Editor ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung zu

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
4. Wählen Sie Bearbeiten aus.
5. Wählen Sie Visual.
6. Wählen Sie im Workflow-Diagramm die Aktion aus, die Ihr benutzerdefiniertes Docker-Image für die Laufzeitumgebung verwenden soll.
7. Wählen Sie die Registerkarte Konfiguration aus.
8. Füllen Sie unten die folgenden Felder aus.

Docker-Image für die Laufzeitumgebung — optional

Geben Sie die Registrierung an, in der Ihr Image gespeichert ist. Gültige Werte sind:

- CODECATALYST(YAML-Editor)

Das Bild wird in der CodeCatalyst Registrierung gespeichert.

- Docker Hub (visueller Editor) oder DockerHub (YAML-Editor)

Das Bild wird in der Docker Hub-Image-Registry gespeichert.

- Andere Registrierung (visueller Editor) oder Other (YAML-Editor)

Das Bild wird in einer benutzerdefinierten Bildregistrierung gespeichert. Jede öffentlich verfügbare Registrierung kann verwendet werden.

- Amazon Elastic Container Registry (visueller Editor) oder ECR (YAML-Editor)

Das Bild wird in einem Image-Repository der Amazon Elastic Container Registry gespeichert. Um ein Bild in einem Amazon ECR-Repository zu verwenden, benötigt diese Aktion Zugriff auf Amazon ECR. Um diesen Zugriff zu aktivieren, müssen Sie eine [IAM-Rolle](#) erstellen, die die folgenden Berechtigungen und eine benutzerdefinierte

Vertrauensrichtlinie umfasst. (Sie können eine bestehende Rolle so ändern, dass sie die Berechtigungen und die Richtlinie einbezieht, wenn Sie möchten.)

Die IAM-Rolle muss die folgenden Berechtigungen in ihrer Rollenrichtlinie enthalten:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

Die IAM-Rolle muss die folgende benutzerdefinierte Vertrauensrichtlinie enthalten:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter [Erstellen einer Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Nachdem Sie die Rolle erstellt haben, müssen Sie sie der Aktion über eine Umgebung zuweisen. Weitere Informationen finden Sie unter [Zuordnen einer Umgebung, einer Kontoverbindung und einer IAM-Rolle zu einer Workflow-Aktion](#).

ECR-Bild-URL, Docker Hub-Bild oder Bild-URL

Geben Sie eines der folgenden Elemente an:

- Wenn Sie eine CODECATALYST Registrierung verwenden, legen Sie für das Image eines der folgenden [aktiven](#) Images fest:
  - 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
- Wenn Sie eine Docker Hub-Registry verwenden, legen Sie für das Image den Namen des Docker Hub-Images und das optionale Tag fest.

Beispiel: `postgres:latest`

- Wenn Sie eine Amazon ECR-Registrierung verwenden, setzen Sie das Bild auf die Amazon ECR-Registrierungs-URI.

Beispiel: `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

- Wenn Sie eine benutzerdefinierte Registrierung verwenden, setzen Sie das Bild auf den Wert, der von der benutzerdefinierten Registrierung erwartet wird.
9. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
  10. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um mit dem YAML-Editor ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung zuzuweisen

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.



2. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
3. Wählen Sie Bearbeiten aus.
4. Wählen Sie YAML.
5. Suchen Sie die Aktion, der Sie ein Docker-Image für die Laufzeitumgebung zuweisen möchten.
6. Fügen Sie in der Aktion einen Container Abschnitt und die zugrunde liegenden Registry Image Eigenschaften hinzu. Weitere Informationen finden Sie in der Beschreibung der Container Registry und den Image Eigenschaften in der [Aktionen](#) Für Ihre Aktion.
7. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
8. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## Beispiele

Die folgenden Beispiele zeigen, wie Sie einer Aktion in der Workflow-Definitionsdatei ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung zuweisen.

## Themen

- [Beispiel: Verwenden eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung, um Unterstützung für Node.js 18 mit Amazon ECR hinzuzufügen](#)
- [Beispiel: Verwenden eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung, um Unterstützung für Node.js 18 mit Docker Hub hinzuzufügen](#)

Beispiel: Verwenden eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung, um Unterstützung für Node.js 18 mit Amazon ECR hinzuzufügen

Das folgende Beispiel zeigt, wie Sie ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung verwenden, um Unterstützung für Node.js 18 mit [Amazon ECR](#) hinzuzufügen.

Configuration:

Container:

Registry: ECR

Image: public.ecr.aws/amazonlinux/amazonlinux:2023

Beispiel: Verwenden eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung, um Unterstützung für Node.js 18 mit Docker Hub hinzuzufügen

[Das folgende Beispiel zeigt, wie Sie ein benutzerdefiniertes Docker-Image für die Laufzeitumgebung verwenden, um Unterstützung für Node.js 18 mit Docker Hub hinzuzufügen.](#)

```
Configuration:
  Container:
    Registry: DockerHub
    Image: node:18.18.2
```

## Arbeiten mit Umgebungen

In einer Umgebung, die nicht mit einer [Entwicklungsumgebung](#) verwechselt werden soll, wird Code bereitgestellt. Sie enthält in der Regel eine Instance einer laufenden Anwendung zusammen mit der zugehörigen Infrastruktur. Sie können Ihrer Umgebung einen Namen wie Entwicklung, Test, Staging oder Produktion geben. Alle Bereitstellungen, die von in einer Umgebung generiert werden CodeCatalyst, werden auf der Seite Umgebungen angezeigt. Um eine Umgebung einzurichten, geben Sie ihr einen Namen, z. B. `my-production-environment`, und verknüpfen ihn dann mit Ihrem AWS-Konto.

Zusätzlich zur Anzeige von Bereitstellungsinformationen dient eine Umgebung auch als Mechanismus, mit dem AWS IAM-Rollen Workflow-[Aktionen](#) zugewiesen werden.

Können mehrere Umgebungen in einem einzigen Workflow vorhanden sein?

Ja. Wenn ein Workflow mehrere Aktionen enthält, kann jeder dieser Aktionen eine Umgebung zugewiesen werden. Sie könnten beispielsweise einen Workflow haben, der zwei Bereitstellungsaktionen umfasst, wobei eine `my-staging-environment` einer Umgebung und eine andere einer `my-production-environment` Umgebung zugewiesen ist.

Welche Aktionen unterstützen Umgebungen?

Die folgenden Aktionen unterstützen die Anzeige ihrer Bereitstellungsinformationen auf der Seite Umgebungen:

- AWS CloudFormation Stack bereitstellen – Weitere Informationen finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#)
- Bereitstellen in Amazon ECS – Weitere Informationen finden Sie unter [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#)

- Im Kubernetes-Cluster bereitstellen – Weitere Informationen finden Sie unter [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#)
- AWS CDK bereitstellen – Weitere Informationen finden Sie unter [Aktion „AWS CDK Deploy“ hinzufügen](#)

### Note

Wenn Sie einer Aktion den Zugriff und die Ausführung von Vorgängen in Ihrem AWS Konto erlauben möchten, müssen Sie sie einer Umgebung zuordnen. Viele Aktionen unterstützen eine Umgebungszuordnung, einschließlich, aber nicht beschränkt auf die zuvor aufgeführten Aktionen. Sie können feststellen, welche Aktionen eine Umgebungszuordnung unterstützen, da sie eine Umgebungs-Dropdown-Liste auf der Registerkarte Konfiguration im [visuellen Editor](#) enthalten.

## Unterstützte Regionen

Auf der Seite Umgebungen können Ressourcen in jeder AWS Region angezeigt werden.

## Sind Umgebungen obligatorisch?

Eine Umgebung ist obligatorisch, wenn die ihr zugewiesene Workflow-Aktion Ressourcen in der AWS Cloud bereitstellt oder aus anderen Gründen mit -AWSServices kommuniziert (z. B. Überwachung und Berichterstellung).

## Erstellen einer Umgebung

Verwenden Sie die folgenden Anweisungen, um eine leere Umgebung zu erstellen, die Sie später einer Workflow-Aktion zuordnen können.

Bevor Sie beginnen

Sie benötigen Folgendes:

- Ein CodeCatalyst Leerzeichen. Weitere Informationen finden Sie unter [Einrichten CodeCatalyst](#).
- Ein CodeCatalyst Projekt. Weitere Informationen finden Sie unter [Ein Projekt mit einem Blueprint erstellen](#).

- Eine AWS Kontoverbindung, die die IAM-Rollen enthält, die Ihre Workflow-Aktion für den Zugriff auf benötigt AWS. Sie können maximal eine Kontoverbindung pro Umgebung verwenden. Weitere Informationen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

#### Note

Sie können eine Umgebung ohne Kontoverbindung erstellen. Sie müssen jedoch zurückkehren und die Verbindung später hinzufügen.

### So erstellen Sie eine Umgebung

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Umgebungen aus.
4. Geben Sie unter Umgebungsname einen Namen ein, z. B. **Production** oder **Staging**.
5. Wählen Sie unter Umgebungstyp eine der folgenden Optionen aus:
  - Nicht produktiv – Eine Umgebung, in der Sie Ihre Anwendung testen können, um sicherzustellen, dass sie wie vorgesehen funktioniert, bevor Sie sie in die Produktion verschieben.
  - Produktion – Eine „Live“-Umgebung, die öffentlich verfügbar ist und Ihre abgeschlossene Anwendung hostet.

Wenn Sie Produktion auswählen, wird in der Benutzeroberfläche neben allen Aktionen, denen die Umgebung zugeordnet ist, ein Produktions-Ausweis angezeigt. Mit dem Abzeichen können Sie schnell sehen, welche Aktionen in der Produktion bereitgestellt werden. Abgesehen vom Auftreten des Abzeichens gibt es keine Unterschiede zwischen Produktions- und Nicht-Produktionsumgebungen.

6. (Optional) Wählen Sie unter VPC-Verbindung eine VPC-Verbindung aus, die Sie dieser Umgebung zuordnen möchten. Weitere Informationen zum Erstellen dieser VPC-Verbindung finden Sie unter [Verwalten von Amazon Virtual Private Clouds](#) im CodeCatalyst - Administratorhandbuch.
7. (Optional) Geben Sie unter Beschreibung eine Beschreibung wie ein **Production environment for the hello-world app**.

8. Wählen Sie unter Verbindung – optional die AWS Kontoverbindung aus, die Sie dieser Umgebung zuordnen möchten. Stellen Sie sicher, dass die Kontoverbindung die IAM-Rollen enthält, die Sie der Umgebung zuordnen möchten. Weitere Informationen zum Erstellen dieser Verbindung finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).
9. Wählen Sie Umgebung erstellen aus. CodeCatalyst Erstellt eine leere Umgebung.

## Nächste Schritte

- Nachdem Sie eine Umgebung erstellt haben, können Sie sie einer Workflow-Aktion zuordnen. Weitere Informationen finden Sie unter [Zuordnen einer Umgebung, einer Kontoverbindung und einer IAM-Rolle zu einer Workflow-Aktion](#).

## Zuordnen einer Umgebung, einer Kontoverbindung und einer IAM-Rolle zu einer Workflow-Aktion

Wenn Sie eine Umgebung, eine Kontoverbindung und eine IAM-Rolle mit einer [unterstützten Workflow-Aktion](#) verknüpfen, steht die IAM-Rolle für die Verwendung durch die Aktion zur Verfügung. Zusätzlich zum Zugriff auf eine IAM-Rolle kann die Aktion auch ihre Bereitstellungsinformationen in die Seite Umgebungen importieren. Weitere Informationen finden Sie unter [Welche Aktionen unterstützen Umgebungen?](#)

Verwenden Sie die folgenden Anweisungen, um eine Umgebung, eine Kontoverbindung und eine IAM-Rolle einer Aktion zuzuordnen.

Schritt 1: Zuordnen der Umgebung, der Kontoverbindung und der Rolle zu einer Workflow-Aktion

Gehen Sie wie folgt vor, um eine Umgebung, eine Kontoverbindung und eine Rolle einer Workflow-Aktion zuzuordnen.

### Visual

So verknüpfen Sie eine Umgebung, eine Kontoverbindung und eine Rolle mit einer Workflow-Aktion mithilfe des visuellen Editors

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.

4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual aus.
7. Wählen Sie im Workflow-Diagramm eine Aktion aus, die von Umgebungen unterstützt wird. Weitere Informationen finden Sie unter [Welche Aktionen unterstützen Umgebungen?](#).
8. Wählen Sie die Registerkarte Konfiguration aus und geben Sie wie folgt Informationen in die Felder ein.

### Umgebung

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Kontoverbindung oder Verbindung – optional (je nachdem, welcher verfügbar ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können maximal eine Kontoverbindung unter angebenEnvironment.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

### Rolle

Geben Sie den Namen der IAM-Rolle an, die diese Aktion verwendet, um auf -AWSServices wie Amazon S3 und Amazon ECR zuzugreifen und diese auszuführen. Stellen Sie sicher, dass diese Rolle zu Ihrer Kontoverbindung hinzugefügt wird. Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

#### Note

Möglicherweise können Sie hier den Namen der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle angeben, sofern sie über ausreichende Berechtigungen

verfügt. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Wenn Sie die Rolle nicht in der Liste sehen, liegt das daran, dass Sie sie nicht der Kontoverbindung zugeordnet haben. Weitere Informationen finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

9. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu validieren.
10. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

So verknüpfen Sie eine Umgebung, eine Kontoverbindung und eine Rolle mit einer Workflow-Aktion mithilfe des YAML-Editors

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML aus.
7. Fügen Sie in der Workflow-Aktion, die Sie einer Umgebung zuordnen möchten, Code hinzu, der dem folgenden ähnelt:

```
action-name
  Environment:
    Name: environment-name
```

**Connections:**

- Name: *account-connection-name*
- Role: *iam-role-name*

Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) für Ihre Aktion.

8. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Schritt 2: Füllen der Umgebung

Nachdem Sie eine Umgebung, eine Kontoverbindung und eine Rolle der Workflow-Aktion zugeordnet haben, können Sie die Seite Umgebungen mit Bereitstellungsinformationen füllen. Verwenden Sie die folgenden Anweisungen, um die Seite Umgebungen auszufüllen.

### Note

Die Seite Umgebungen wird nur von einer Teilmenge von Workflow-Aktionen unterstützt. Weitere Informationen finden Sie unter [Welche Aktionen unterstützen Umgebungen?](#).

## So füllen Sie die Umgebung

1. Wenn eine Workflow-Ausführung nicht automatisch gestartet wurde, als Sie Ihre Änderungen in festgeschrieben haben [Schritt 1: Zuordnen der Umgebung, der Kontoverbindung und der Rolle zu einer Workflow-Aktion](#), starten Sie eine Ausführung manuell wie folgt:
  - a. Wählen Sie im Navigationsbereich CI/CD und anschließend Workflows aus.
  - b. Wählen Sie den Namen des Workflows aus, in dem Sie eine Ausführung starten möchten. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
  - c. Wählen Sie Ausführen aus.

Die Workflow-Ausführung startet eine neue Bereitstellung, wodurch Ihre Anwendungsressourceninformationen unter Umgebungen CodeCatalyst hinzugefügt.

2. Stellen Sie sicher, dass Ihre Anwendungsressourcen in Ihrer Umgebung angezeigt werden:



- a. Wählen Sie im Navigationsbereich CI/CD und dann Umgebungen aus.
- b. Wählen Sie Ihre Umgebung aus (z. B. `Production`).
- c. Wählen Sie die Registerkarte Bereitstellungsaktivität und überprüfen Sie, ob eine Bereitstellung mit dem Status `SUCCEEDED` angezeigt wird. Dies deutet darauf hin, dass eine Workflow-Ausführung Ihre Anwendungsressourcen erfolgreich bereitgestellt hat.
- d. Wählen Sie die Registerkarte Bereitstellungsziele und überprüfen Sie, ob Ihre Anwendungsressourcen angezeigt werden.

## Verwalten einer Umgebung

Gehen Sie wie folgt vor, um eine Umgebung zu verwalten, indem Sie sie einer VPC-Verbindung oder einem zuordnenAWS-Konto.

### Zuordnen einer VPC-Verbindung zu einer Umgebung

Wenn eine Aktion mit einer Umgebung konfiguriert ist, die über eine VPC-Verbindung verfügt, wird die Aktion unter Einhaltung der Netzwerkregeln und Zugriffsressourcen ausgeführt, die von der zugehörigen VPC angegeben werden. Die gleiche VPC-Verbindung kann von einer oder mehreren Umgebungen verwendet werden.

Verwenden Sie die folgenden Anweisungen, um eine VPC-Verbindung mit einer Umgebung zu verknüpfen.

### So verknüpfen Sie eine VPC-Verbindung mit einer Umgebung

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Umgebungen aus.
4. Wählen Sie Ihre Umgebung aus (z. B. `Production`).
5. Wählen Sie die Registerkarte Umgebungseigenschaften aus.
6. Wählen Sie VPC-Verbindung verwalten, wählen Sie Ihre gewünschte VPC-Verbindung und wählen Sie Bestätigen aus. Dadurch wird Ihre ausgewählte VPC-Verbindung mit dieser Umgebung verknüpft.

Weitere Informationen finden Sie unter [Verwalten von Amazon Virtual Private Clouds](#) im CodeCatalyst Administratorhandbuch für .

## Zuordnen eines AWS-Konto zu einer Umgebung

Verwenden Sie die folgenden Anweisungen, um einen AWS-Konto mit einer Umgebung zu verknüpfen.

So verknüpfen Sie einen AWS-Konto mit einer Umgebung

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Umgebungen aus.
4. Wählen Sie Ihre Umgebung aus (z. B. Production).
5. Wählen Sie die Registerkarte Umgebungseigenschaften aus.
6. Wählen Sie Zuordnen, AWS-Kontowählen Sie Ihr AWS-Kontogewünschtes und wählen Sie Zuordnen aus. Dadurch wird Ihr ausgewählter AWS-Konto mit dieser Umgebung verknüpft.

Weitere Informationen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

## Arbeiten mit Datei-Caching

Wenn das Zwischenspeichern von Dateien aktiviert ist, speichern die Build- und Testaktionen Dateien auf der Festplatte in einem Cache und stellen sie in nachfolgenden Workflow-Ausführungen aus diesem Cache wieder her. Durch das Zwischenspeichern wird die Latenz reduziert, die durch das Erstellen oder Herunterladen von Abhängigkeiten verursacht wird, die sich zwischen den Läufen nicht geändert haben. CodeCatalyst unterstützt auch Fallback-Caches, mit denen Teil-Caches wiederhergestellt werden können, die einige der benötigten Abhängigkeiten enthalten. Dies trägt dazu bei, die Latenzauswirkungen eines Cache-Fehlers zu reduzieren.

### Note

Datei-Caching ist nur mit den CodeCatalyst [Build](#) - und [Testaktionen](#) von Amazon verfügbar und nur, wenn sie für die Verwendung des [EC2-Compute-Typs](#) konfiguriert sind.

### Themen

- [Über das Zwischenspeichern von Dateien](#)
- [Einen Cache erstellen](#)

- [Beschränkungen](#)

## Über das Zwischenspeichern von Dateien

Mit dem Datei-Caching können Sie Ihre Daten in mehreren Caches organisieren, auf die jeweils unter der Eigenschaft verwiesen wird. `FileCaching` Jeder Cache speichert ein Verzeichnis, das durch einen bestimmten Pfad angegeben ist. Das angegebene Verzeichnis wird bei future Workflow-Ausführungen wiederhergestellt. Im Folgenden finden Sie ein Beispiel für ein YAML-Snippet für das Caching mit mehreren Caches mit dem Namen und. `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 verwendet mehrschichtiges Caching, das aus einem lokalen Cache und einem Remote-Cache besteht. Wenn bereitgestellte Flotten oder On-Demand-Computer auf einen Cache-Fehler in einem lokalen Cache stoßen, werden Abhängigkeiten aus einem Remote-Cache wiederhergestellt. Aus diesem Grund kann es bei einigen Aktionsläufen zu Latenzen beim Herunterladen eines Remote-Caches kommen.

CodeCatalyst wendet Cache-Zugriffsbeschränkungen an, um sicherzustellen, dass eine Aktion in einem Workflow die Caches eines anderen Workflows nicht ändern kann. Dadurch wird jeder Workflow vor anderen Workflows geschützt, die möglicherweise falsche Daten weiterleiten, die sich auf Builds oder Bereitstellungen auswirken. Einschränkungen werden durch Cache-Bereiche durchgesetzt, die Caches für jeden Workflow und jede Branch-Paarung isolieren. Beispielsweise `feature-A` hat ein Branch einen anderen Datei-Cache als ein Schwester-Branch. `workflow-A` `workflow-A` `feature-B`

Cachefehler treten auf, wenn ein Workflow nach einem bestimmten Dateicache sucht und ihn nicht finden kann. Dies kann mehrere Gründe haben, z. B. wenn ein neuer Branch erstellt wird oder wenn auf einen neuen Cache verwiesen wird, der noch nicht erstellt wurde. Es kann auch vorkommen, wenn ein Cache abläuft, was standardmäßig 14 Tage nach seiner letzten Verwendung der Fall ist. CodeCatalyst unterstützt Fallback-Caches, um Cache-Fehlschläge zu vermeiden und die Rate von Cache-Treffern zu erhöhen. Fallback-Caches sind alternative Caches und bieten die Möglichkeit, partielle Caches wiederherzustellen, bei denen es sich um eine ältere Version eines Caches handeln kann. Ein Cache wird wiederhergestellt, indem zunächst unter dem Eigenschaftsnamen nach einem Treffer gesucht wird. `FileCaching` Falls er nicht gefunden wird, wird er ausgewertet. `RestoreKeys` Wenn sowohl für den Eigenschaftsnamen als auch für die gesamte Eigenschaft ein Cache-Fehler auftritt `RestoreKeys`, wird der Workflow weiter ausgeführt, da das Zwischenspeichern nach bestem Bemühen erfolgt und nicht garantiert werden kann.

## Einen Cache erstellen

Sie können die folgenden Anweisungen verwenden, um Ihrem Workflow einen Cache hinzuzufügen.

### Visual

Um einen Cache mit dem Visual Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.

7. Wählen Sie im Workflow-Diagramm die Aktion aus, zu der Sie Ihren Cache hinzufügen möchten.
8. Wählen Sie Konfiguration.
9. Wählen Sie unter Datei-Caching — optional die Option Cache hinzufügen aus und geben Sie Informationen wie folgt in die Felder ein:

#### Key (Schlüssel)

Geben Sie den Namen Ihrer primären Cache-Eigenschaft an. Die Namen der Cache-Eigenschaften müssen innerhalb Ihres Workflows eindeutig sein. Jede Aktion kann bis zu fünf Einträge enthalten `FileCaching`.

#### Pfad

Geben Sie den zugehörigen Pfad für Ihren Cache an.

#### Schlüssel wiederherstellen — optional

Geben Sie den Wiederherstellungsschlüssel an, der als Fallback verwendet werden soll, wenn die primäre Cache-Eigenschaft nicht gefunden werden kann. Die Namen der Wiederherstellungsschlüssel müssen innerhalb Ihres Workflows eindeutig sein. Jeder Cache kann bis zu fünf Einträge enthalten `RestoreKeys`.

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein, und wählen Sie dann erneut Commit aus.

## YAML

Um einen Cache mit dem YAML-Editor hinzuzufügen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.

5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer Workflow-Aktion Code hinzu, der dem folgenden ähnelt:

```
action-name:
  Configuration:
    Steps: ...
  Caching:
    FileCaching:
      key-name:
        Path: file-path
        # # Specify any additional fallback caches
        # RestoreKeys:
        # - restore-key
```

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Festschreiben zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Beschränkungen

Im Folgenden sind die Einschränkungen für den Eigenschaftsnamen und `RestoreKeys`:

- Namen müssen innerhalb eines Workflows eindeutig sein.
- Namen sind auf alphanumerische Zeichen (A-Z, a-z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt.
- Namen können bis zu 180 Zeichen lang sein.
- Jede Aktion kann bis zu fünf Caches enthalten. `FileCaching`
- Jeder Cache kann bis zu fünf Einträge enthalten. `RestoreKeys`

Im Folgenden sind die Einschränkungen für Pfade aufgeführt:

- Sternchen (\*) sind nicht erlaubt.
- Pfade können bis zu 255 Zeichen lang sein.

## Mit Paketen arbeiten

Ein Paket ist ein Paket, das sowohl Software als auch die Metadaten enthält, die für die Installation der Software und die Auflösung aller Abhängigkeiten erforderlich sind. CodeCatalyst unterstützt das npm-Paketformat.

Ein Paket besteht aus:

- Ein Name (webpackist zum Beispiel der Name eines beliebigen npm-Pakets)
- Ein optionaler [Namespace](#) (zum Beispiel in) @types @types/node
- Eine Reihe von [Versionen](#) (zum Beispiel, 1.0.0 1.0.1, 1.0.2)
- Metadaten auf Paketebene (z. B. npm dist-Tags)

In CodeCatalyst können Sie Pakete in Paket-Repositorys veröffentlichen und Pakete aus CodeCatalyst Paket-Repositorys in Ihren Workflows verwenden. Sie können eine Build- oder Testaktion mit einem CodeCatalyst Paket-Repository konfigurieren, um den npm-Client einer Aktion automatisch so zu konfigurieren, dass Pakete aus dem angegebenen Repository gesendet und abgerufen werden.

Weitere Informationen zu Paketen finden Sie unter [Pakete in CodeCatalyst](#).

### Note

Derzeit unterstützen Build- und Testaktionen CodeCatalyst Paket-Repositorys.

Themen

- [Verwendung von CodeCatalyst Paket-Repositorys in Workflows](#)
- [Beispiele für die Verwendung von Paketen in Workflows](#)

## Verwendung von CodeCatalyst Paket-Repositorys in Workflows

CodeCatalystIn können Sie Ihren Build- und Testaktionen in Ihrem Workflow ein CodeCatalyst Paket-Repository hinzufügen. Ihr Paket-Repository muss mit einem Paketformat wie npm konfiguriert sein. Sie können sich auch dafür entscheiden, eine Abfolge von Bereichen für Ihr ausgewähltes Paket-Repository einzubeziehen.

Verwenden Sie die folgenden Anweisungen, um eine Paketkonfiguration anzugeben, die mit einer Workflow-Aktion verwendet werden soll.

## Visual

So geben Sie die Paketkonfiguration an, die eine Aktion verwenden soll (visueller Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, mit der Sie ein Paket-Repository konfigurieren möchten.
8. Wählen Sie Pakete aus.
9. Wählen Sie im Dropdownmenü Konfiguration hinzufügen die Paketkonfiguration aus, die Sie für Ihre Workflow-Aktionen verwenden möchten.
10. Wählen Sie Paket-Repository hinzufügen aus.
11. Geben Sie im Dropdownmenü Paket-Repository den Namen Ihres CodeCatalyst Paket-Repositorys an, das die Aktion verwenden soll.

Weitere Hinweise zu Paket-Repositorys finden Sie unter. [Paket-Repositorys](#)

12. (Optional) Geben Sie im Feld Bereiche — optional eine Abfolge von Bereichen an, die Sie in Ihrer Paketregistrierung definieren möchten.

Bei der Definition von Bereichen wird das angegebene Paket-Repository als Registrierung für alle aufgelisteten Bereiche konfiguriert. Wenn ein Paket mit dem Bereich über den npm-Client angefordert wird, verwendet es dieses Repository anstelle des Standard-Repositorys. Jedem Bereichsnamen muss ein „@“ vorangestellt werden.

Wenn dieser Scopes Wert weggelassen wird, wird das angegebene Paket-Repository als Standardregistrierung für alle von der Aktion verwendeten Pakete konfiguriert.



Weitere Informationen zu Bereichen finden Sie unter [Paket-Namespaces](#) und Pakete mit [Geltungsbereich](#).

13. Wählen Sie Add (Hinzufügen) aus.
14. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
15. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um die Paketkonfiguration anzugeben, die eine Aktion verwenden soll (YAML-Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer Aktion Code hinzu, der dem folgenden ähnelt:

```
action-name:
  Configuration:
    Packages:
      NpmConfiguration:
        PackageRegistries:
          - PackagesRepository: package-repository
        Scopes:
          - "@scope"
```

Weitere Informationen finden Sie in der Beschreibung der Packages Eigenschaft unter [Aktionsreferenz zum Erstellen und Testen](#) Für Ihre Aktion.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.

9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Beispiele für die Verwendung von Paketen in Workflows

Die folgenden Beispiele zeigen, wie Sie in der Workflow-Definitionsdatei auf Pakete verweisen.

### Themen

- [Beispiel: Pakete definieren mit NpmConfiguration](#)
- [Beispiel: Überschreiben der Standardregistrierung](#)
- [Beispiel: Bereiche in Ihrer Paketregistrierung überschreiben](#)

### Beispiel: Pakete definieren mit **NpmConfiguration**

Das folgende Beispiel zeigt, wie Sie ein Paket mit `NpmConfiguration` in Ihrer Workflow-Definitionsdatei definieren.

```
Actions:
  Build:
    Identifier: aws/build-beta@v1
    Configuration:
      Packages:
        NpmConfiguration:
          PackageRegistries:
            - PackagesRepository: main-repo
            - PackagesRepository: scoped-repo
          Scopes:
            - "@scope1"
```

In diesem Beispiel wird der `npm`-Client wie folgt konfiguriert:

```
default: main-repo
@scope1: scoped-repo
```

In diesem Beispiel sind zwei `Repository`s definiert. Die Standardregistrierung ist so festgelegt, `main-repo` wie sie ohne Bereich definiert ist. Der Bereich `@scope1` ist in `PackageRegistries` für konfiguriert `scoped-repo`.

## Beispiel: Überschreiben der Standardregistrierung

Das folgende Beispiel zeigt Ihnen, wie Sie die Standardregistrierung überschreiben können.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-repo-1
    - PackagesRepository: my-repo-2
    - PackagesRepository: my-repo-3
```

In diesem Beispiel wird der npm-Client wie folgt konfiguriert:

```
default: my-repo-3
```

Wenn Sie mehrere Standard-Repositorys angeben, hat das letzte Repository Priorität. In diesem Beispiel ist das zuletzt aufgeführte Repository `my-repo-3`, was bedeutet, dass npm eine Verbindung herstellen wird. `my-repo-3` Dies überschreibt die `my-repo-1` Repositorys und `my-repo-2`

## Beispiel: Bereiche in Ihrer Paketregistrierung überschreiben

Das folgende Beispiel zeigt Ihnen, wie Sie einen Bereich in Ihrer Paketregistrierung überschreiben können.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-default-repo
    - PackagesRepository: my-repo-1
  Scopes:
    - "@scope1"
    - "@scope2"
  - PackagesRepository: my-repo-2
  Scopes:
    - "@scope2"
```

In diesem Beispiel wird der npm-Client wie folgt konfiguriert:

```
default: my-default-repo
@scope1: my-repo-1
@scope2: my-repo-2
```

Wenn Sie übergeordnete Bereiche einbeziehen, hat das letzte Repository Priorität. In diesem Beispiel wurde dieser Bereich `@scope2` zum letzten Mal für konfiguriert. `PackageRegistries my-repo-2`. Dadurch wird der für `@scope2 my-repo-1` konfigurierte Bereich außer Kraft gesetzt.

## Mit Läufen arbeiten

Ein Lauf ist eine einzelne Iteration eines Workflows. CodeCatalyst führt während eines Laufs die in der Workflow-Konfigurationsdatei definierten Aktionen aus und gibt die zugehörigen Protokolle, Artefakte und Variablen aus.

### Themen

- [Einen Workflow-Lauf starten](#)
- [Anhalten einer Workflow-Ausführung](#)
- [Anzeigen des Status und der Details der Workflow-Ausführung](#)
- [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#)

## Einen Workflow-Lauf starten

Gehen Sie wie folgt vor, um eine Workflow-Ausführung manuell zu starten.

### Note

Sie können eine Workflow-Ausführung auch automatisch starten, indem Sie [einen Trigger konfigurieren](#).

Um einen Workflow manuell zu starten, führen Sie ihn manuell aus

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen des Workflows aus, den Sie ausführen möchten. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Run (Ausführen) aus.

## Anhalten einer Workflow-Ausführung

Gehen Sie wie folgt vor, um eine laufende Workflow-Ausführung zu beenden. Möglicherweise möchten Sie einen Lauf beenden, wenn er versehentlich gestartet wurde.

Wenn Sie eine Workflow-Ausführung beenden, CodeCatalyst wartet, bis laufende Aktionen abgeschlossen sind, bevor die Ausführung in der CodeCatalyst Konsole als Beendet markiert wird. Alle Aktionen, die nicht gestartet werden konnten, werden nicht gestartet und als „Abgebrochen“ markiert.

### Note

Befindet sich ein Lauf in der Warteschlange (d. h., es gibt keine laufenden Aktionen), wird der Lauf sofort gestoppt.

Um eine Workflow-Ausführung zu beenden

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie unter Workflows die Option Läufe aus und wählen Sie die Ausführung, die gerade ausgeführt wird, aus der Liste aus.
5. Wählen Sie Beenden aus.

## Anzeigen des Status und der Details der Workflow-Ausführung

Sie können den Status und die Details einer einzelnen Workflow-Ausführung oder mehrerer Ausführungen gleichzeitig anzeigen.

### Note

Sie können auch den Workflow-Status anzeigen, der sich vom Workflow-Ausführungsgsstatus unterscheidet. Weitere Informationen finden Sie unter [Anzeigen des Workflow-Status](#).

## Themen

- [Status der Workflow-Ausführung](#)
- [Anzeigen des Status und der Details einer einzelnen Ausführung](#)
- [Anzeigen des Status und der Details aller Ausführungen in Ihrem Projekt](#)
- [Anzeigen des Status und der Details aller Ausführungen eines bestimmten Workflows](#)
- [Anzeigen von Ausführungen eines Workflows im Workflow-Diagramm](#)

## Status der Workflow-Ausführung

Eine Workflow-Ausführung kann einen der folgenden Status haben:

- **Erfolgreich** – Die Workflow-Ausführung wurde erfolgreich verarbeitet.
- **Fehlgeschlagen** – Eine oder mehrere Aktionen in der Workflow-Ausführung sind fehlgeschlagen.
- **In Bearbeitung** – Die Workflow-Ausführung wird gerade verarbeitet.
- **Gestoppt** – Eine Person hat die Workflow-Ausführung gestoppt, während sie in Bearbeitung war.
- **Stoppen** – Die Workflow-Ausführung wird derzeit gestoppt.
- **Abgebrochen** – Die Workflow-Ausführung wurde von abgebrochen, CodeCatalyst da der zugehörige Workflow gelöscht oder aktualisiert wurde, während die Ausführung ausgeführt wurde.
- **Ersetzt** – Tritt nur auf, wenn Sie den [eretzten Ausführungsmodus](#) konfiguriert haben. Die Workflow-Ausführung wurde von abgebrochen, CodeCatalyst da sie durch eine spätere Workflow-Ausführung ersetzt wurde.

## Anzeigen des Status und der Details einer einzelnen Ausführung

Möglicherweise möchten Sie den Status und die Details einer einzelnen Workflow-Ausführung anzeigen, um zu überprüfen, ob sie erfolgreich war, um zu sehen, zu welchem Zeitpunkt sie abgeschlossen wurde oder um zu sehen, wer oder was sie gestartet hat.

So zeigen Sie den Status und die Details einer einzelnen Ausführung an

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und anschließend Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.

5. Wählen Sie unter dem Namen des Workflows die Option Ausführen aus.
6. Wählen Sie unter Ausführungsverlauf in der Spalte Ausführungs-ID eine Ausführung aus.  
Beispiel: Run-95a4d
7. Führen Sie unter dem Namen der Ausführung einen der folgenden Schritte aus:
  - Visualisierung, um ein Workflow-Diagramm zu sehen, das die Aktionen Ihrer Workflow-Ausführung und ihren Status zeigt (siehe [Status der Workflow-Ausführung](#)). Diese Ansicht zeigt auch das Quell-Repository und den Zweig, die während der Ausführung verwendet wurden.

Wählen Sie im Workflow-Diagramm eine Aktion aus, um Details wie Protokolle, Berichte und Ausgaben anzuzeigen, die während der Ausführung von der Aktion generiert wurden. Die angezeigten Informationen hängen davon ab, welcher Aktionstyp ausgewählt ist. Weitere Informationen zum Anzeigen von Build- oder Bereitstellungsprotokollen finden Sie unter [Ergebnisse einer Build-Aktion anzeigen](#) oder [Bereitstellungsprotokolle anzeigen](#).

- YAML, um die Workflow-Definitionsdatei anzuzeigen, die für die Ausführung verwendet wurde.
- Artefakte, um die Artefakte anzuzeigen, die durch die Workflow-Ausführung erzeugt wurden. Weitere Informationen zu Artefakten finden Sie unter [Mit Artefakten arbeiten](#).
- Berichte, um die Testberichte und andere Arten von Berichten anzuzeigen, die durch die Workflow-Ausführung erstellt wurden. Weitere Informationen zu Berichten finden Sie unter [Typen von Testberichten](#).
- Variablen zum Anzeigen der Ausgabevariablen, die durch die Workflow-Ausführung erzeugt wurden. Weitere Informationen zu Variablen finden Sie unter [Arbeiten mit Variablen](#).

#### Note

Wenn der übergeordnete Workflow der Ausführung gelöscht wurde, wird oben auf der Seite mit den Ausführungsdetails eine Meldung angezeigt, die auf diese Tatsache hinweist.

## Anzeigen des Status und der Details aller Ausführungen in Ihrem Projekt

Möglicherweise möchten Sie den Status und die Details aller Workflow-Ausführungen in Ihrem Projekt anzeigen, um zu verstehen, wie viele Workflow-Aktivitäten in Ihrem Projekt ausgeführt werden, und mehr über den Gesamtzustand Ihrer Workflows erfahren.


So zeigen Sie den Status und die Details aller Ausführungen in Ihrem Projekt an

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie unter Workflows die Option Ausführen aus.

Alle Ausführungen für alle Workflows in allen Verzweigungen über alle Repositorys in Ihrem Projekt werden angezeigt.

Die Seite enthält die folgenden Spalten:

- Ausführungs-ID – Die eindeutige Kennung der Ausführung. Wählen Sie den Link Ausführungs-ID, um detaillierte Informationen über die Ausführung anzuzeigen.
- Status – Der Verarbeitungsstatus der Workflow-Ausführung. Weitere Informationen zum Ausführungsstatus finden Sie unter [Status der Workflow-Ausführung](#).
- Auslöser – Die Person, der Commit, die Pull-Anforderung (PR) oder der Zeitplan, die/der die Workflow-Ausführung gestartet hat. Weitere Informationen finden Sie unter [Arbeiten mit Auslösern](#).
- Workflow – Der Name des Workflows, für den eine Ausführung gestartet wurde, sowie das Quell-Repository und die Verzweigung, in der sich die Workflow-Definitionsdatei befindet. Möglicherweise müssen Sie die Spaltenbreite erweitern, um diese Informationen anzuzeigen.

 Note

Wenn diese Spalte auf Nicht verfügbar gesetzt ist, liegt dies normalerweise daran, dass der zugehörige Workflow gelöscht oder verschoben wurde.

- Startzeit – Die Uhrzeit, zu der die Workflow-Ausführung gestartet wurde.
- Dauer – Wie lange die Ausführung des Workflows gedauert hat. Eine sehr lange oder sehr kurze Dauer kann auf Probleme hinweisen.
- Endzeit – Die Zeit, zu der die Workflow-Ausführung beendet wurde.

Anzeigen des Status und der Details aller Ausführungen eines bestimmten Workflows

Möglicherweise möchten Sie den Status und die Details aller Läufe anzeigen, die einem bestimmten Workflow zugeordnet sind, um festzustellen, ob Läufe Engpässe innerhalb des Workflows



verursachen oder um zu sehen, welche Läufe derzeit ausgeführt werden oder abgeschlossen wurden.

So zeigen Sie den Status und die Details aller Ausführungen eines bestimmten Workflows an

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie unter dem Namen des Workflows die Option Ausführen aus.

Die Ausführungen, die dem ausgewählten Workflow zugeordnet sind, werden angezeigt.

Die Seite ist in zwei Abschnitte unterteilt:


- Aktive Ausführungen – Zeigt laufende Ausführungen an. Diese Ausführungen befinden sich in einem der folgenden Zustände: In Bearbeitung.
- Ausführungsverlauf – Zeigt abgeschlossene Ausführungen an (d. h. nicht in Bearbeitung).

Weitere Informationen zu Ausführungsstatus finden Sie unter [Status der Workflow-Ausführung](#).

## Anzeigen von Ausführungen eines Workflows im Workflow-Diagramm


Sie können den Status aller Ausführungen eines Workflows anzeigen, während sie den Workflow gemeinsam durchlaufen. Die Ausführungen werden im Workflow-Diagramm angezeigt (im Gegensatz zu in einer Listenansicht). Auf diese Weise erhalten Sie eine visuelle Darstellung, welche Ausführungen von welchen Aktionen verarbeitet werden und welche Ausführungen in einer Warteschlange warten.

So zeigen Sie den Status mehrerer Ausführungen an, während sie gemeinsam durch einen Workflow laufen

 Note

Dieses Verfahren gilt nur, wenn Ihr Workflow den Ausführungsmodus in der Warteschlange oder ersetzt verwendet. Weitere Informationen finden Sie unter [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#).

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen des Workflows aus, der die Ausführungen enthält, die Sie anzeigen möchten. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.

 Note

Stellen Sie sicher, dass Sie sich eine Workflow-Seite und keine Ausführungsseite ansehen.

5. Wählen Sie oben links die Registerkarte Letzter Status aus.

Ein Workflow-Diagramm wird angezeigt.

6. Überprüfen Sie das Workflow-Diagramm. Das Diagramm zeigt alle Ausführungen, die derzeit innerhalb des Workflows ausgeführt werden, sowie die letzten Ausführungen, die abgeschlossen wurden. Genauer gesagt:
  - Ausführungen, die oben vor den Quellen angezeigt werden, werden in die Warteschlange gestellt und warten darauf, zu starten.
  - Ausführungen, die zwischen Aktionen angezeigt werden, werden in die Warteschlange gestellt und warten darauf, von der nächsten Aktion verarbeitet zu werden.
  - Ausführungen, die in einer Aktion vorkommen, sind 1. wird derzeit von der Aktion verarbeitet, 2. hat die Verarbeitung durch die Aktion abgeschlossen, oder 3. wurde nicht von der Aktion verarbeitet (in der Regel weil eine vorherige Aktion fehlgeschlagen ist).

## Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen

Wenn mehrere Workflow-Ausführungen gleichzeitig ausgeführt werden, werden sie standardmäßig in eine CodeCatalyst Warteschlange gestellt und nacheinander in der Reihenfolge verarbeitet, in der sie gestartet wurden. Sie können dieses Standardverhalten ändern, indem Sie einen Ausführungsmodus angeben. Es gibt einige Ausführungsmodi:

- (Standard) Ausführungsmodus in Warteschlange — CodeCatalyst Prozesse werden nacheinander ausgeführt
- Abgelöster Ausführungsmodus — CodeCatalyst Prozesse werden nacheinander ausgeführt, wobei neuere Läufe ältere überholen
- Parallellauffunktion — CodeCatalyst Prozesse laufen parallel

### Themen

- [Über den Ausführungsmodus in der Warteschlange](#)
- [Über den abgelösten Ausführungsmodus](#)
- [Über den Parallellaufmodus](#)
- [Den Ausführungsmodus ändern](#)

### Über den Ausführungsmodus in der Warteschlange

Im Modus „Ausführung in der Warteschlange“ werden Läufe nacheinander ausgeführt, wobei Warteläufe eine Warteschlange bilden.

Warteschlangen bilden sich an den Eintrittspunkten zu Aktionen und Aktionsgruppen, sodass Sie mehrere Warteschlangen innerhalb desselben Workflows haben können (siehe [Figure 1](#)). Wenn ein Lauf in der Warteschlange in eine Aktion aufgenommen wird, ist die Aktion gesperrt und es können keine weiteren Läufe mehr hinzugefügt werden. Wenn der Lauf abgeschlossen ist und die Aktion beendet wird, wird die Aktion entsperrt und ist bereit für den nächsten Lauf.

[Figure 1](#) veranschaulicht einen Workflow, der im Ausführungsmodus in der Warteschlange konfiguriert ist. Es zeigt:

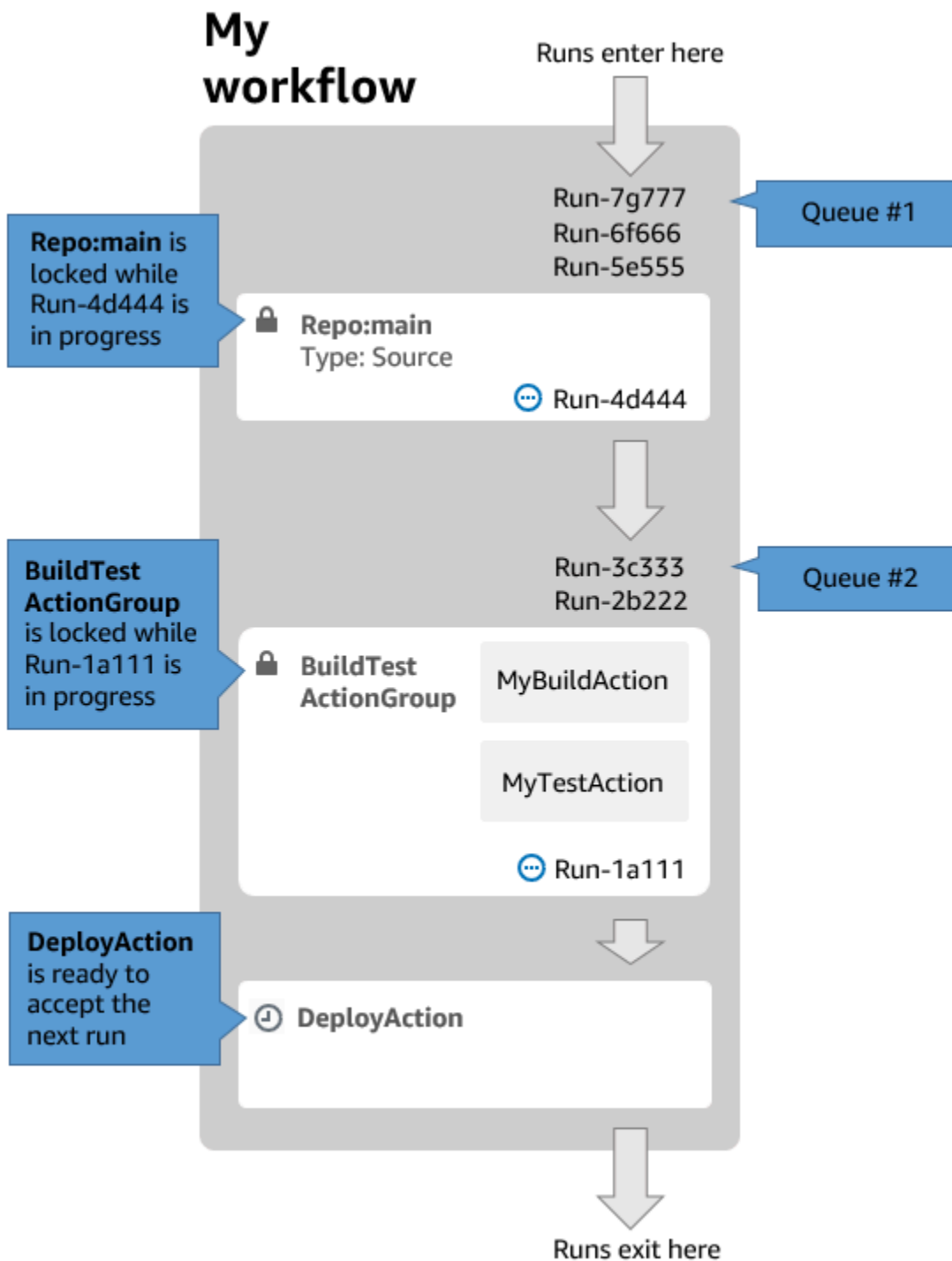
- Sieben Läufe arbeiten sich ihren Weg durch den Workflow.
- Zwei Warteschlangen: eine außerhalb des Eingangs zur Eingabequelle (repo:Main) und eine außerhalb des Eingangs zur Aktion. BuildTestActionGroup

- Zwei gesperrte Blöcke: die Eingangsquelle (repo:Main) und der BuildTestActionGroup

So werden sich die Dinge entwickeln, wenn der Workflow läuft und die Verarbeitung abgeschlossen ist:

- Wenn Run-4D444 das Klonen des Quell-Repositorys abgeschlossen hat, verlässt es die Eingabequelle und tritt der Warteschlange hinter Run-3c333 bei. Dann wird Run-5e555 die Eingabequelle aufrufen.
- Wenn Run-1a111 mit dem Erstellen und Testen fertig ist, wird die Aktion beendet und die Aktion gestartet. BuildTestActionGroupDeployAction Dann wird Run-2b222 die Aktion starten. BuildTestActionGroup

Abbildung 1: Ein Workflow, der im Modus „Ausführung in der Warteschlange“ konfiguriert ist



Verwenden Sie den Ausführungsmodus in der Warteschlange, wenn:

- Sie möchten eine one-to-one Beziehung zwischen Features und Läufen beibehalten — diese Features können gruppiert werden, wenn der Modus „Ersetzt“ verwendet wird. Wenn Sie beispielsweise Feature 1 in Commit 1 zusammenführen, wird Lauf 1 gestartet, und wenn Sie Feature 2 in Commit 2 zusammenführen, startet Lauf 2 usw. Wenn Sie den ersetzten Modus

anstelle des Warteschlangenmodus verwenden würden, werden Ihre Features (und Commits) in der Ausführung gruppiert, die die anderen ersetzt.

- Sie möchten Rennbedingungen und unerwartete Probleme vermeiden, die bei der Verwendung des Parallelmodus auftreten können. Wenn beispielsweise zwei Softwareentwickler, Wang und Saanvi, Workflow-Läufe ungefähr zur gleichen Zeit starten, um sie auf einem Amazon ECS-Cluster bereitzustellen, kann Wangs Lauf Integrationstests auf dem Cluster beginnen, während der Lauf von Saanvi neuen Anwendungscode für den Cluster bereitstellt, was dazu führt, dass Wangs Tests entweder fehlschlagen oder den falschen Code testen. Ein weiteres Beispiel könnte sein, dass Sie ein Ziel haben, das keinen Sperrmechanismus hat. In diesem Fall könnten die beiden Läufe die Änderungen des jeweils anderen auf unerwartete Weise überschreiben.
- Sie möchten die Belastung der Rechenressourcen begrenzen, die für die Verarbeitung Ihrer Läufe CodeCatalyst verwendet werden. Wenn Sie beispielsweise drei Aktionen in Ihrem Workflow haben, können Sie maximal drei Läufe gleichzeitig ausführen lassen. Wenn die Anzahl der Durchläufe, die gleichzeitig ausgeführt werden können, begrenzt wird, lässt sich der Durchsatz der Durchläufe besser vorhersagen.
- Sie möchten die Anzahl der Anfragen einschränken, die der Workflow an Dienste von Drittanbietern stellt. Ihr Workflow könnte beispielsweise eine Build-Aktion enthalten, die Anweisungen zum Abrufen eines Images aus Docker Hub enthält. [Docker Hub begrenzt die Anzahl der Pull-Anfragen](#), die Sie stellen können, auf eine bestimmte Anzahl pro Stunde und Konto. Wenn Sie das Limit überschreiten, werden Sie gesperrt. Wenn Sie den Ausführungsmodus in der Warteschlange verwenden, um Ihren Durchsatz zu verlangsamen, werden weniger Anfragen an Docker Hub pro Stunde generiert, wodurch das Potenzial für Sperrungen und daraus resultierende Build- und Run-Fehler begrenzt wird.

Maximale Warteschlangengröße: 50

Hinweise zur maximalen Warteschlangengröße:

- Die maximale Warteschlangengröße bezieht sich auf die maximale Anzahl von Durchläufen, die in allen Warteschlangen im Workflow zulässig sind.
- Wenn eine Warteschlange länger als 50 Läufe wird CodeCatalyst, werden die 51. und die nachfolgenden Läufe gelöscht.

Verhalten bei Fehlern:

Wenn ein Lauf nicht mehr reagiert, während er durch eine Aktion verarbeitet wird, werden die dahinter stehenden Läufe in der Warteschlange zurückgehalten, bis die Aktion das Timeout erreicht. Aktionen laufen nach einer Stunde ab.

Wenn ein Lauf innerhalb einer Aktion fehlschlägt, darf der erste Lauf, der sich dahinter in der Warteschlange befindet, fortgesetzt werden.

### Über den abgelösten Ausführungsmodus

Der abgelöste Ausführungsmodus entspricht dem Ausführungsmodus in der Warteschlange, mit der Ausnahme, dass:

- Wenn ein Lauf in der Warteschlange einen anderen Lauf in der Warteschlange einholt, ersetzt (übernimmt) der spätere Lauf den früheren Lauf, und der frühere Lauf wird abgebrochen und als „ersetzt“ markiert.
- Aufgrund des im ersten Punkt beschriebenen Verhaltens kann eine Warteschlange nur einen Lauf enthalten, wenn der Modus „Ersetzte Ausführung“ verwendet wird.

Wenn Sie den Arbeitsablauf [Figure 1](#) als Leitfaden verwenden, würde die Anwendung des ersetzten Ausführungsmodus auf diesen Workflow zu folgenden Ergebnissen führen:

- Run-7g777 würde die anderen beiden Läufe in der Warteschlange ersetzen und wäre der einzige in Warteschlange #1 verbleibende Lauf. Run-6F666 und Run-5E555 würden abgebrochen.
- Run-3c333 würde Run-2b222 ersetzen und wäre der einzige verbleibende Lauf in Warteschlange #2. Run-2b222 würde abgebrochen werden.

Verwenden Sie den abgelösten Ausführungsmodus, wenn Sie:

- besserer Durchsatz als im Warteschlangenmodus
- noch weniger Anfragen an Dienste von Drittanbietern als im Warteschlangenmodus; dies ist vorteilhaft, wenn für den Drittanbieter-Service Ratenbegrenzungen gelten, wie z. B. Docker Hub

### Über den Parallellaufmodus

Im Parallellaufmodus sind Läufe unabhängig voneinander und warten nicht, bis andere Läufe abgeschlossen sind, bevor sie gestartet werden. Es gibt keine Warteschlangen, und der Durchsatz der Ausführung wird nur dadurch begrenzt, wie schnell die Aktionen innerhalb des Workflows abgeschlossen werden.

Verwenden Sie den parallel Ausführungsmodus in Entwicklungsumgebungen, in denen jeder Benutzer seinen eigenen Feature-Branch hat und auf Zielen bereitgestellt wird, die nicht von anderen Benutzern gemeinsam genutzt werden.

### Important

Wenn Sie ein gemeinsames Ziel haben, für das mehrere Benutzer bereitstellen können, z. B. eine Lambda-Funktion in einer Produktionsumgebung, verwenden Sie den Parallelmodus nicht, da dies zu Rennbedingungen führen kann. Eine Race-Condition tritt auf, wenn parallel Workflow-Läufe versuchen, eine gemeinsam genutzte Ressource gleichzeitig zu ändern, was zu unvorhersehbaren Ergebnissen führt.

Maximale Anzahl parallel Läufe: 1000 pro CodeCatalyst Feld

Den Ausführungsmodus ändern

Sie können den Ausführungsmodus auf „Warteschlange“, „ersetzt“ oder „Parallel“ setzen. Die Standardeinstellung ist in der Warteschlange.

Wenn Sie den Ausführungsmodus von „In Warteschlange“ oder „ersetzt“ in „Parallel“ ändern, werden die Läufe in der Warteschlange CodeCatalyst abgebrochen und die Läufe, die derzeit von einer Aktion verarbeitet werden, beendet, bevor sie abgebrochen werden.

Wenn Sie den Ausführungsmodus von „Parallel“ in „In Warteschlange“ oder „ersetzt“ ändern, werden alle derzeit laufenden parallel CodeCatalyst Läufe abgeschlossen. Alle Läufe, die Sie starten, nachdem Sie den Ausführungsmodus in Warteschlange oder ersetzt geändert haben, verwenden den neuen Modus.

Visual

Um den Ausführungsmodus mit dem Visual Editor zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.



5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie oben rechts die Option Workflow-Eigenschaften aus.
7. Erweitern Sie Erweitert und wählen Sie unter Ausführungsmodus eine der folgenden Optionen aus:
  - a. In der Warteschlange — siehe [Über den Ausführungsmodus in der Warteschlange](#)
  - b. Ersetzt — siehe [Über den abgelösten Ausführungsmodus](#)
  - c. Parallel — siehe [Über den Parallelaufmodus](#)
8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit.

## YAML

Um den Ausführungsmodus mit dem YAML-Editor zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie die RunMode Eigenschaft wie folgt hinzu:

```
Name: Workflow_6d39
SchemaVersion: "1.0"
RunMode: QUEUED|SUPERSEDED|PARALLEL
```

Weitere Informationen finden Sie in der Beschreibung der RunMode Immobilie im [Immobilien auf höchstem Niveau](#) Abschnitt der [Referenz zur Workflow-Definition](#).

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.

9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Arbeiten mit Secrets

Es kann vorkommen, dass Sie sensible Daten, wie z. B. Authentifizierungsanmeldeinformationen, in Ihren Workflows verwenden müssen. Das Speichern dieser Werte im Klartext an einer beliebigen Stelle in Ihrem Repository sollte vermieden werden, da jeder, der Zugriff auf das Repository hat, das das Secret enthält, sie sehen kann. Ebenso sollten diese Werte nicht direkt in Workflow-Definitionen verwendet werden, da sie als Dateien in Ihrem Repository sichtbar sind. Mit können Sie diese Werte schützen CodeCatalyst, indem Sie Ihrem Projekt ein Secret hinzufügen und dann in Ihrer Workflow-Definitionsdatei auf das Secret verweisen. Beachten Sie, dass Sie maximal fünf Secrets pro Aktion haben können.

### Note

Secrets können nur verwendet werden, um Passwörter und sensible Informationen in der Workflow-Definitionsdatei zu ersetzen.

### Themen

- [Erstellen eines Secrets](#)
- [Bearbeiten eines Secrets](#)
- [Verwenden eines Secrets](#)
- [Löschen eines Secrets](#)

## Erstellen eines Secrets

Gehen Sie wie folgt vor, um ein Secret zu erstellen. Das Secret enthält die sensiblen Informationen, die Sie vor der Ansicht ausblenden möchten.

### Note

Secrets sind für -Aktionen sichtbar und werden nicht maskiert, wenn sie in eine Datei geschrieben werden.

## So erstellen Sie ein Secret

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Secrets aus.
3. Wählen Sie Create secret (Secret erstellen) aus.
4. Geben Sie die folgenden Informationen ein:

### Name

Geben Sie einen Namen für Ihr Secret ein.

### Wert

Geben Sie den Wert für das Secret ein. Dies sind die sensiblen Informationen, die Sie vor der Ansicht ausblenden möchten. Standardmäßig wird der Wert nicht angezeigt. Um den Wert anzuzeigen, wählen Sie Wert anzeigen aus.

### Beschreibung

(Optional) Geben Sie eine Beschreibung für Ihr Secret ein.

5. Wählen Sie Erstellen.

## Bearbeiten eines Secrets

Gehen Sie wie folgt vor, um ein Secret zu bearbeiten.

### So bearbeiten Sie ein Secret

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Secrets aus.
3. Wählen Sie in der Liste der Secrets das Secret aus, das Sie bearbeiten möchten.
4. Wählen Sie Bearbeiten aus.
5. Bearbeiten Sie die folgenden Eigenschaften:

### Wert

Geben Sie den Wert für das Secret ein. Dies ist der Wert, den Sie vor der Ansicht ausblenden möchten. Standardmäßig wird der Wert nicht angezeigt.

## Beschreibung

(Optional) Geben Sie eine Beschreibung für Ihr Secret ein.

6. Wählen Sie Speichern.

## Verwenden eines Secrets

Um ein Secret in einer Workflow-Aktion zu verwenden, müssen Sie die Referenz-ID des Secrets abrufen und diese ID in der Workflow-Aktion verwenden.

### Themen

- [Abrufen der Kennung eines Secrets](#)
- [Verweisen auf ein Secret in einem Workflow](#)

### Abrufen der Kennung eines Secrets

Gehen Sie wie folgt vor, um die Referenz-ID des Secrets abzurufen. Sie fügen diese Kennung zu Ihrem Workflow hinzu.

So rufen Sie die Referenz-ID des Secrets ab

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Secrets aus.
3. Suchen Sie in der Liste der Secrets das Secret, das Sie verwenden möchten.
4. Kopieren Sie in der Spalte Referenz-ID die Kennung des Secrets. Im Folgenden finden Sie die Syntax für die Referenz-ID :

```
${Secrets.<name>}
```

### Verweisen auf ein Secret in einem Workflow

Gehen Sie wie folgt vor, um auf ein Secret in einem Workflow zu verweisen.

So verweisen Sie auf ein Secret

1. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.

2. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
3. Wählen Sie Bearbeiten aus.
4. Wählen Sie YAML aus.
5. Ändern Sie die YAML, um die Kennung des Secrets zu verwenden. Um beispielsweise einen Benutzernamen und ein Passwort zu verwenden, die mit dem `curl` Befehl als Secrets gespeichert werden, verwenden Sie einen Run Befehl ähnlich dem folgenden:

```
- Run: curl -u <username-secret-identifizier>:<password-secret-identifizier> https://example.com
```

6. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
7. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Löschen eines Secrets

Gehen Sie wie folgt vor, um ein Secret und die Secret-Referenz-ID zu löschen.

### Note

Bevor Sie ein Secret löschen, empfehlen wir, die Referenz-ID des Secrets aus allen Workflow-Aktionen zu entfernen. Wenn Sie das Secret löschen, ohne die Referenz-ID zu löschen, schlägt die Aktion bei der nächsten Ausführung fehl.

So löschen Sie die Referenz-ID eines Secrets aus einem Workflow

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
3. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
4. Wählen Sie Bearbeiten aus.

5. Wählen Sie YAML aus.
6. Suchen Sie im Workflow nach der folgenden Zeichenfolge:

```
`${Secrets}.
```

Dadurch werden alle Referenz-IDs aller Secrets gefunden.

7. Löschen Sie die Referenz-ID des ausgewählten Secrets oder ersetzen Sie sie durch einen Klartextwert.
8. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu validieren.
9. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

So löschen Sie ein Geheimnis

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich CI/CD und dann Secrets aus.
3. Wählen Sie in der Liste der Secrets das Secret aus, das Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Geben Sie **eindelete**, um den Löschvorgang zu bestätigen.
6. Wählen Sie Löschen.

## Mit Quellen arbeiten

Eine Quelle, auch Eingabequelle genannt, ist ein Quell-Repository, auf das eine [Workflow-Aktion](#) Zugriff benötigt, um ihre Aufgaben auszuführen. Beispielsweise muss eine Workflow-Aktion möglicherweise auf eine Quelle zugreifen, um Komponententests abzurufen und sie anhand der Quelldateien Ihrer Anwendung auszuführen.

CodeCatalyst Workflows unterstützen die folgenden Quellen:

- CodeCatalyst Quell-Repositorys — Weitere Informationen finden Sie unter [Quell-Repositoryn in CodeCatalyst](#).
- GitHub Quell-Repositoryn — Weitere Informationen finden Sie unter [Verwendung von GitHub Repositorys in CodeCatalyst](#)

## Themen

- [Angabe der Quelle, in der die Workflow-Definitionsdatei gespeichert werden soll](#)
- [Angabe der Quelle, die eine Workflow-Aktion verwenden soll](#)
- [Verweisen auf Dateien in einem Quell-Repository](#)
- [Von der Quelle erzeugte Variablen](#)

### Angabe der Quelle, in der die Workflow-Definitionsdatei gespeichert werden soll

Verwenden Sie die folgenden Anweisungen, um das CodeCatalyst Quell-Repository anzugeben, in dem Sie Ihre Workflow-Definitionsdatei speichern möchten. Wenn Sie lieber ein GitHub Quell-Repository angeben möchten, finden Sie stattdessen weitere Informationen unter [Verwendung von GitHub Repositories in CodeCatalyst](#).

Das Quell-Repository, in dem sich Ihre Workflow-Definitionsdatei befindet, ist durch die Bezeichnung, `WorkflowSource` gekennzeichnet.

#### Note

Sie geben das Quell-Repository an, in dem sich Ihre Workflow-Definitionsdatei befindet, wenn Sie Ihre Workflow-Definitionsdatei zum ersten Mal übertragen. Nach diesem Commit sind das Repository und die Workflow-Definitionsdatei dauerhaft miteinander verknüpft. Die einzige Möglichkeit, das Repository nach dem ersten Commit zu ändern, besteht darin, den Workflow in einem anderen Repository neu zu erstellen.

Um das Quell-Repository anzugeben, in dem die Workflow-Definitionsdatei gespeichert werden soll

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie Workflow erstellen und erstellen Sie den Workflow. Weitere Informationen finden Sie unter [Um einen Workflow mit dem Visual Editor zu erstellen](#).

Während der Workflow-Erstellung werden Sie aufgefordert, das CodeCatalyst Repository anzugeben, in dem Sie Ihre Workflow-Definitionsdatei speichern möchten.

## Angabe der Quelle, die eine Workflow-Aktion verwenden soll

Verwenden Sie die folgenden Anweisungen, um ein Quell-Repository anzugeben, das mit einer Workflow-Aktion verwendet werden soll. Beim Start bündelt die Aktion die Dateien im konfigurierten Quell-Repository zu einem Artefakt, lädt das Artefakt in das [Docker-Image der Laufzeitumgebung](#) herunter, in dem die Aktion ausgeführt wird, und schließt dann die Verarbeitung mit den heruntergeladenen Dateien ab.

### Note

Derzeit können Sie innerhalb einer Workflow-Aktion nur ein Quell-Repository angeben, nämlich das Quell-Repository, in dem sich die Workflow-Definitionsdatei befindet (im Verzeichnis). `.codecatalyst/workflows/` Dieses Quell-Repository wird durch das Label `WorkflowSource` dargestellt.

## Visual

Um das Quell-Repository anzugeben, das eine Aktion verwenden soll (visueller Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, für die Sie die Quelle angeben möchten.
8. Wählen Sie Eingaben.
9. Gehen Sie unter Quellen — optional wie folgt vor:

Geben Sie die Labels an, die die Quell-Repositorys repräsentieren, die für die Aktion benötigt werden. Derzeit wird nur die Bezeichnung, die das Quell-Repository darstellt `WorkflowSource`, in dem Ihre Workflow-Definitionsdatei gespeichert ist, unterstützt.



Wenn Sie eine Quelle weglassen, müssen Sie mindestens ein Eingabeartefakt unter angeben. *action-name*/Inputs/Artifacts

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um das Quell-Repository anzugeben, das eine Aktion verwenden soll (YAML-Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer Aktion Code hinzu, der dem folgenden ähnelt:

```
action-name:
  Inputs:
    Sources:
      - WorkflowSource
```

Weitere Informationen finden Sie in der Beschreibung der Sources Eigenschaft unter [Referenz zur Workflow-Definition](#) Für Ihre Aktion.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Verweisen auf Dateien in einem Quell-Repository

Wenn Sie Dateien haben, die sich in einem Quell-Repository befinden, und Sie in einer Ihrer Workflow-Aktionen auf diese Dateien verweisen müssen, gehen Sie wie folgt vor.

### Note

Weitere Informationen finden Sie auch unter [Referenzieren von Dateien in einem Artefakt](#).

Um auf eine Datei in einem Quell-Repository zu verweisen

- Fügen Sie in der Aktion, in der Sie auf eine Datei verweisen möchten, Code hinzu, der dem folgenden ähnelt:

```
Actions:
  My-action:
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - run: cd my-app && cat file1.jar
```

Im vorherigen Code sucht die Aktion im Verzeichnis im my-app Stammverzeichnis des WorkflowSource Quell-Repositorys nach der file1.jar Datei und zeigt sie an.

## Von der Quelle erzeugte Variablen

Wenn ein Workflow ausgeführt wird, erzeugt seine Quelle Variablen, die Sie in nachfolgenden Workflow-Aktionen verwenden können. Einzelheiten finden Sie [Variablen BranchName "" und CommitId ""](#) in der [Liste der vordefinierten Variablen](#).

## Arbeiten mit Auslösern

Mit einem Workflow-Auslöser oder einfach einem Auslöser können Sie eine Workflow-Ausführung automatisch starten, wenn bestimmte Ereignisse auftreten, z. B. ein Code-Push. Möglicherweise möchten Sie Auslöser konfigurieren, damit Ihre Softwareentwickler Workflow-Ausführungen manuell über die CodeCatalyst Konsole starten können.

Sie können drei Arten von Auslösern verwenden:

- Push – Ein Code-Push-Auslöser bewirkt, dass eine Workflow-Ausführung gestartet wird, wenn ein Commit übertragen wird.
- Pull-Anforderung – Ein Pull-Anforderungsauslöser führt dazu, dass eine Workflow-Ausführung gestartet wird, wenn eine Pull-Anforderung entweder erstellt, überarbeitet oder geschlossen wird.
- Zeitplan – Ein Zeitplanauslöser bewirkt, dass eine Workflow-Ausführung nach einem von Ihnen definierten Zeitplan gestartet wird. Erwägen Sie, einen Zeitplanauslöser zu verwenden, um jede Nacht Builds Ihrer Software auszuführen, damit Ihre Softwareentwickler am nächsten Tag arbeiten können.

Sie können Push-, Pull-Anforderungs- und Zeitplanauslöser allein oder in Kombination im selben Workflow verwenden.

#### Tip

Um einen Auslöser in Aktion zu sehen, starten Sie ein Projekt mit einer Vorlage. Die meisten Vorlagen enthalten einen Workflow mit einem Auslöser. Suchen Sie in der Workflow-Definitionsdatei des Blueprints nach der `Trigger` Eigenschaft. Weitere Informationen über Pläne finden Sie unter [Ein Projekt mit einem Blueprint erstellen](#).

## Themen

- [Eine gemeinsame Auslöserkonfiguration](#)
- [Überlegungen zu Auslösern bei der Verzweigung](#)
- [Hinzufügen eines Push-, Pull- oder Zeitplanauslösers](#)
- [Beispiele für Auslöser](#)

## Eine gemeinsame Auslöserkonfiguration

In diesem Abschnitt wird beschrieben, wie Sie Auslöser für eine gängige Software-Release- und Verzweigungsstrategie einrichten.

Software-Release- und Verzweigungsstrategie:

- Sie haben Anwendungscode in einem Quell-Repository.

- Ihre `main` Verzweigung enthält finalisierten Code, der immer bereit für die Veröffentlichung ist.
- Ihre Softwareentwickler nehmen ihre Änderungen an Feature-Zweigen außerhalb des `main` Zweigs vor.
- Ihre Softwareentwickler [erstellen eine Pull-Anfrage](#), in der sie aufgefordert werden, ihren Feature-Zweig zusammenzuführen `main`, wenn ihr Feature bereit ist.

Sie möchten, dass diese Pull-Anforderung automatisch einen Workflow startet, der die Anwendung mithilfe der Dateien auf dem Feature-Zweig des Softwareentwicklers erstellt und testet – aber nicht bereitstellt.

- Ihre Softwareentwickler überprüfen den Build und die Tests, um sicherzustellen, dass jede Menge gut aussieht. Anschließend [führen sie die Pull-Anforderung](#) in der `main` Verzweigung zusammen.

Sie möchten, dass die Zusammenführung automatisch einen Workflow startet, der Ihren Anwendungscode erstellt und bereitstellt.

Vorgeschlagene Workflow-/Auslöserkonfiguration:

Aufgrund der zuvor beschriebenen Softwareverzweigungsstrategie sollten Sie zwei Workflows verwenden:

- Workflow 1 erstellt und testet Ihre Anwendung, wenn eine Pull-Anforderung erstellt oder überarbeitet wird.
- Workflow 2 erstellt und stellt Ihre Anwendung bereit, wenn eine Pull-Anforderung zusammengeführt wird.

Workflow 1 würde wie folgt aussehen:

Triggers:

- Type: PULLREQUEST

Branches:

- `main`

Events:

- OPEN
- REVISION

Actions:

BuildAction:

*instructions-for-building-the-app*

TestAction:

### *instructions-for-test-the-app*

Der vorherige Auslösercode startet automatisch eine Workflow-Ausführung, wenn ein Softwareentwickler eine Pull-Anforderung erstellt (oder [eine ändert](#)), in der er aufgefordert wird, seinen Feature-Zweig mit dem main Zweig zusammenzuführen. CodeCatalyst startet eine Workflow-Ausführung mit dem Code in der Quellverzweigung (d. h. der Feature-Verzweigung des Entwicklers). Der Workflow erstellt und stellt die Anwendung bereit.

Workflow 2 würde wie folgt aussehen:

```
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildAction:
    instructions-for-building-the-app
  DeployAction:
    instructions-for-deploying-the-app
```

Wenn im vorherigen Auslösercode eine Zusammenführung mit main erfolgt, wird der PUSH Auslöser aktiviert. CodeCatalyst startet eine Workflow-Ausführung mit dem Code in der main Verzweigung (der jetzt den Code aus der Pull-Anforderung enthält). Der Workflow erstellt und stellt die Anwendung bereit.

Anweisungen zum Hinzufügen von Auslösern zu einer Workflow-Definitionsdatei finden Sie unter [Hinzufügen eines Push-, Pull- oder Zeitplanauslösers](#).

Weitere Beispiele für Auslöser und zusätzliche Erklärungen finden Sie unter [Beispiele für Auslöser](#).

## Überlegungen zu Auslösern bei der Verzweigung

In diesem Abschnitt werden einige der wichtigsten Überlegungen beim Einrichten von Auslösern beschrieben, die Verzweigungen enthalten.

- Überlegungen 1: Wenn Sie sowohl bei Push- als auch bei Pull-Anforderungsauslösern einen Zweig angeben möchten, müssen Sie den Zielzweig (oder „an“) in der Auslöserkonfiguration angeben. Geben Sie niemals den Quellzweig (oder „von“) an.

Im folgenden Beispiel main aktiviert ein Push von einer beliebigen Verzweigung zu den Workflow.

```
Triggers:
- Type: PUSH
  Branches:
    - main
```

Im folgenden Beispiel `main` aktiviert eine Pull-Anforderung von einer beliebigen Verzweigung in den Workflow.

```
Triggers:
- Type: PULLREQUEST
  Branches:
    - main
  Events:
    - OPEN
    - REVISION
```

- Überlegungen 2: Bei Push-Auslösern wird der Workflow nach der Aktivierung des Workflows mit der Workflow-Definitionsdatei und den Quelldateien im Zielzweig ausgeführt.
- Überlegungen 3: Bei Pull-Anforderungsauslösern wird der Workflow nach der Aktivierung des Workflows mit der Workflow-Definitionsdatei und den Quelldateien im Quellzweig ausgeführt (obwohl Sie den Zielzweig in der Auslöserkonfiguration angegeben haben).
- Überlegungen 4: Der genaue Auslöser in einer Verzweigung wird möglicherweise nicht in einer anderen Verzweigung ausgeführt.

Betrachten Sie den folgenden Push-Auslöser:

```
Triggers:
- Type: PUSH
  Branches:
    - main
```

Wenn die Workflow-Definitionsdatei, die diesen Auslöser enthält, in vorhanden ist `main` und in geklont wird `test`, beginnt der Workflow niemals automatisch mit der Verwendung der Dateien in `test` (obwohl Sie den Workflow manuell starten können, wenn Sie die Dateien in verwendet `test`). Lesen Sie die Überlegungen 1 und 2, um zu verstehen, warum der Workflow niemals automatisch mit den Dateien in ausgeführt wird `test`.

Berücksichtigen Sie auch den folgenden Auslöser für Pull-Anfragen:

**Triggers:**

- Type: PULLREQUEST

**Branches:**

- main

**Events:**

- OPEN
- REVISION

Wenn die Workflow-Definitionsdatei, die diesen Auslöser enthält, in vorhanden ist `main`, wird der Workflow niemals mit den Dateien in `main` ausgeführt. (Wenn Sie jedoch eine `test` Verzweigung außerhalb von `main` erstellen, wird der Workflow mit den Dateien in `test` ausgeführt.) Lesen Sie die Überlegungen 1 und 3, um zu verstehen, warum.

## Hinzufügen eines Push-, Pull- oder Zeitplanauslösers

Verwenden Sie die folgenden Anweisungen, um Ihrem Workflow einen Push-, Pull- oder Zeitplanauslöser hinzuzufügen.

### Visual

So fügen Sie einen Auslöser hinzu (visueller Editor)

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Namen des Quell-Repositorys oder der Verzweigung filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual aus.
7. Wählen Sie im Workflow-Diagramm das Feld Quelle und Auslöser aus.
8. Wählen Sie im Konfigurationsbereich Add trigger aus.
9. Geben Sie im Dialogfeld Auslöser hinzufügen wie folgt Informationen in die Felder ein.

### Auslösertyp

Geben Sie den Typ des Auslösers an. Sie können einen der folgenden Werte verwenden:

- Push (visueller Editor) oder PUSH (YAML-Editor)

Ein Push-Auslöser startet eine Workflow-Ausführung, wenn eine Änderung in Ihr Quell-Repository übertragen wird. Die Workflow-Ausführung verwendet die Dateien in der Verzweigung, an die Sie übertragen (d. h. die Zielverzweigung).

- Pull-Anforderung (visueller Editor) oder PULLREQUEST (YAML-Editor)

Ein Pull-Anforderungsauslöser startet eine Workflow-Ausführung, wenn eine Pull-Anforderung in Ihrem Quell-Repository geöffnet, aktualisiert oder geschlossen wird. Die Workflow-Ausführung verwendet die Dateien in der Verzweigung, aus der Sie abrufen (d. h. der Quellzweig).

- Zeitplan (visueller Editor) oder SCHEDULE (YAML-Editor)

Ein Zeitplanauslöser startet Workflow-Ausführungen nach einem Zeitplan, der durch einen von Ihnen angegebenen Cron-Ausdruck definiert wird. Für jede Verzweigung in Ihrem Quell-Repository wird mithilfe der Dateien der Verzweigung eine separate Workflow-Ausführung gestartet. (Um die Verzweigungen einzuschränken, für die der Auslöser aktiviert wird, verwenden Sie das Feld Verzweigungen (visueller Editor) oder die Branches Eigenschaft (YAML-Editor).)

Befolgen Sie beim Konfigurieren eines Zeitplanauslösers die folgenden Richtlinien:

- Verwenden Sie nur einen Zeitplanauslöser pro Workflow.
- Wenn Sie mehrere Workflows in Ihrem CodeCatalyst Bereich definiert haben, empfehlen wir Ihnen, nicht mehr als 10 davon gleichzeitig zu starten.
- Stellen Sie sicher, dass Sie den Cron-Ausdruck des Auslösers mit ausreichender Zeit zwischen Ausführungen konfigurieren. Weitere Informationen finden Sie unter [Expression](#).

Beispiele finden Sie unter [Beispiele für Auslöser](#).

### Ereignisse für Pull-Anforderung

Dieses Feld wird nur angezeigt, wenn Sie den Auslösertyp Pull request ausgewählt haben.

Geben Sie den Typ der Pull-Anforderungsereignisse an, die eine Workflow-Ausführung starten. Die folgenden Werte sind gültig:



- Pull-Anforderung wird erstellt (visueller Editor) oder OPEN (YAML-Editor)

Die Workflow-Ausführung wird gestartet, wenn eine Pull-Anforderung erstellt wird.

- Pull-Anforderung wird geschlossen (visueller Editor) oder CLOSED (YAML-Editor)

Die Workflow-Ausführung wird gestartet, wenn eine Pull-Anforderung geschlossen wird. Das Verhalten des CLOSED Ereignisses ist schwierig und wird am besten anhand eines Beispiels verstanden. Weitere Informationen finden Sie unter [Beispiel: Ein Auslöser mit einem Pull-, Verzweig- und einem „CLOSED“-Ereignis](#).

- Eine neue Revision wird vorgenommen, um die Anforderung abzurufen (visueller Editor) oder REVISION (YAML-Editor)

Die Workflow-Ausführung wird gestartet, wenn eine Revision einer Pull-Anforderung erstellt wird. Die erste Revision wird erstellt, wenn die Pull-Anforderung erstellt wird. Danach wird jedes Mal eine neue Revision erstellt, wenn jemand einen neuen Commit an die in der Pull-Anforderung angegebene Quellverzweigung überträgt. Wenn Sie das REVISION Ereignis in Ihren Pull-Anforderungsauslöser aufnehmen, können Sie das OPEN Ereignis weglassen, da eine Obermenge von REVISION ist OPEN.

Sie können mehrere Ereignisse in demselben Pull-Anforderungsauslöser angeben.

Beispiele finden Sie unter [Beispiele für Auslöser](#).

## Plan

Dieses Feld wird nur angezeigt, wenn Sie den Auslösertyp Zeitplan ausgewählt haben.

Geben Sie den Cron-Ausdruck an, der beschreibt, wann Ihr geplanter Workflow ausgeführt werden soll.

Cron-Ausdrücke in CodeCatalyst verwenden die folgende sechsfeldige Syntax, wobei jedes Feld durch ein Leerzeichen getrennt ist:

*Minuten Stunden days-of-month im Monat days-of-week Jahr*

Beispiele für Cron-Ausdrücke

| Minuten | Stunden | Tage des Monats | Monat | Wochentage | Jahr | Bedeutung  |
|---------|---------|-----------------|-------|------------|------|--|
| 0       | 0       | ?               | *     | MO-FR      | *    | Führt jeden Montag bis Freitag einen Workflow um Mitternacht (UTC +0) aus. |
| 0       | 2       | *               | *     | ?          | *    | Führt jeden Tag einen Workflow um 2:00 Uhr (UTC +0) aus.                   |
| 15      | 22      | *               | *     | ?          | *    | Führt jeden Tag einen Workflow um 22:15 Uhr (UTC +0) aus.                  |

| Minuten | Stunden | Tage des Monats | Monat | Wochentage | Jahr      | Bedeutung  |
|---------|---------|-----------------|-------|------------|-----------|--|
| 0/30    | 22-2    | ?               | *     | SAT-SUN    | *         | Führt jeden 30-minütigen Workflow von Samstag bis Sonntag zwischen 22:00 Uhr am Starttag und 2:00 Uhr am darauffolgenden Tag (UTC +0) aus. |
| 45      | 13      | L               | *     | ?          | 2023-2027 | Führt einen Workflow um 13:45 Uhr (UTC +0) am letzten Tag des Monats zwischen den Jahren 2023 und 2027 aus.                                |

Befolgen Sie beim Angeben von Cron-Ausdrücken in CodeCatalyst die folgenden Richtlinien:

- Geben Sie einen einzelnen Cron-Ausdruck pro SCHEDULE Auslöser an.
- Fügen Sie den Cron-Ausdruck in doppelte Anführungszeichen (") im YAML-Editor ein.
- Geben Sie die Zeit in UTC (Coordinated Universal Time) an. Andere Zeitzonen werden nicht unterstützt.
- Konfigurieren Sie mindestens 30 Minuten zwischen Ausführungen. Ein schnellerer Takt wird nicht unterstützt.
- Geben Sie das *days-of-week* Feld *days-of-month* oder an, aber nicht beides. Wenn Sie einen Wert oder ein Sternchen (\*) in einem der Felder angeben, müssen Sie ein Fragezeichen (?) im anderen verwenden. Das Sternchen steht für „alle“ und das Fragezeichen für „alle“.

Weitere Beispiele für Cron-Ausdrücke und Informationen zu Platzhaltern wie ?, \* und L finden Sie in der [Referenz zu Cron-Ausdrücken](#) im Amazon-EventBridge Benutzerhandbuch. Cron-Ausdrücke in EventBridge und CodeCatalyst funktionieren genauso.

Beispiele für Zeitplanauslöser finden Sie unter [Beispiele für Auslöser](#).

### Verzweigungen und Verzweigungsmuster

(Optional)

Geben Sie die Verzweigungen in Ihrem Quell-Repository an, die der Auslöser überwacht, um zu wissen, wann eine Workflow-Ausführung gestartet werden soll. Sie können Regex-Muster verwenden, um Ihre Verzweigungsnamen zu definieren. Verwenden Sie beispielsweise `main.*` um alle Verzweigungen abzugleichen, die mit `main` beginnen.

Die anzugebenden Verzweigungen unterscheiden sich je nach Auslösertyp:

- Geben Sie für einen Push-Auslöser die Verzweigungen an, die Sie an übertragen, d. h. die Zielverzweigungen. Eine Workflow-Ausführung wird pro übereinstimmender Verzweigung gestartet, wobei die Dateien in der übereinstimmenden Verzweigung verwendet werden.

Beispiele: `main.*`, `mainline`

- Geben Sie für einen Pull-Anforderungsauslöser die Verzweigungen an, die Sie an übertragen, d. h. die Zielverzweigungen. Eine Workflow-Ausführung wird pro übereinstimmender Verzweigung gestartet, wobei die Workflow-Definitionsdatei und die Quelldateien in der Quellverzweigung (nicht die übereinstimmende Verzweigung) verwendet werden.

Beispiele: `main.*`, `mainline`, `v1\-.*` (abgleicht Verzweigungen, die mit `beginnev1-`)

- Geben Sie für einen Zeitplanauslöser die Verzweigungen an, die die Dateien enthalten, die Ihre geplante Ausführung verwenden soll. Eine Workflow-Ausführung wird pro übereinstimmender Verzweigung gestartet, wobei die Workflow-Definitionsdatei und die Quelldateien in der übereinstimmenden Verzweigung verwendet werden.

Beispiele: `main.*`, `version\-1\.*`

#### Note

Wenn Sie keine Verzweigungen angeben, überwacht der Auslöser alle Verzweigungen in Ihrem Quell-Repository und startet eine Workflow-Ausführung mit der Workflow-Definitionsdatei und den Quelldateien in:

- Die Verzweigung, an die Sie pushen (für Push-Auslöser). Weitere Informationen finden Sie unter [Beispiel: Ein einfacher Code-Push-Auslöser](#).
- Die Verzweigung, aus der Sie abrufen (für Pull-Anforderungsauslöser). Weitere Informationen finden Sie unter [Beispiel: Ein einfacher Pull-Anforderungsauslöser](#).
- Alle Verzweigungen (für Zeitplanauslöser). Eine Workflow-Ausführung wird pro Verzweigung in Ihrem Quell-Repository gestartet. Weitere Informationen finden Sie unter [Beispiel: Ein einfacher Zeitplanauslöser](#).

Weitere Informationen zu Verzweigungen und Auslösern finden Sie unter [Überlegungen zu Auslösern bei der Verzweigung](#).

Weitere Beispiele finden Sie unter [Beispiele für Auslöser](#).

Geänderte Dateien

Dieses Feld wird nur angezeigt, wenn Sie den Auslösertyp Push -oder Pull-Anforderung ausgewählt haben.

Geben Sie die Dateien oder Ordner in Ihrem Quell-Repository an, die der Auslöser überwacht, um zu wissen, wann eine Workflow-Ausführung gestartet werden soll. Sie können reguläre Ausdrücke verwenden, um Dateinamen oder Pfade abzugleichen.

Beispiele finden Sie unter [Beispiele für Auslöser](#).

10. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu validieren.
11. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

So fügen Sie einen Auslöser hinzu (YAML-Editor)

1. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows aus. Sie können nach dem Quell-Repository oder dem Verzweigungsnamen filtern, in dem der Workflow definiert ist, oder nach Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML aus.
7. Fügen Sie einen `Triggers` Abschnitt und die zugrunde liegenden Eigenschaften hinzu, indem Sie das folgende Beispiel als Leitfaden verwenden. Weitere Informationen finden Sie unter [Triggers](#) im [Referenz zur Workflow-Definition](#).

Ein Code-Push-Auslöser könnte wie folgt aussehen:

```
Triggers:
  - Type: PUSH
    Branches:
      - main
```

Ein Pull-Anforderungsauslöser könnte wie folgt aussehen:

```
Triggers:  
- Type: PULLREQUEST  
  Branches:  
    - main.*  
  Events:  
    - OPEN  
    - REVISION  
    - CLOSED
```

Ein Zeitplanauslöser könnte wie folgt aussehen:

```
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"
```

Weitere Beispiele für Cron-Ausdrücke, die Sie in der `-Expression`Eigenschaft verwenden können, finden Sie unter [Expression](#).

Weitere Beispiele für Push-, Pull-Anforderungs- und Zeitplanauslöser finden Sie unter [Beispiele für Auslöser](#).

8. (Optional) Wählen Sie Validieren, um den YAML-Code des Workflows vor dem Commit zu validieren.
9. Wählen Sie Commit aus, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Beispiele für Auslöser

Die folgenden Beispiele zeigen, wie Sie verschiedene Arten von Auslösern in der Workflow-Definitionsdatei hinzufügen.

### Themen

- [Beispiel: Ein einfacher Code-Push-Auslöser](#)
- [Beispiel: Ein einfacher „Push to Main“-Auslöser](#)

- [Beispiel: Ein einfacher Pull-Anforderungsauslöser](#)
- [Beispiel: Ein einfacher Zeitplanauslöser](#)
- [Beispiel: Ein Auslöser mit einem Zeitplan und Verzweigungen](#)
- [Beispiel: Ein Auslöser mit einem Zeitplan, einem Push und Verzweigungen](#)
- [Beispiel: Ein Auslöser mit einem Pull und Verzweigungen](#)
- [Beispiel: Ein Auslöser mit einem Pull-, Verzweig- und einem „CLOSED“-Ereignis](#)
- [Beispiel: Ein Auslöser mit Push, Verzweigungen und Dateien](#)

### Beispiel: Ein einfacher Code-Push-Auslöser

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung startet, wenn Code an eine Verzweigung in Ihrem Quell-Repository übertragen wird.

Wenn dieser Auslöser aktiviert ist, CodeCatalyst startet eine Workflow-Ausführung mit den Dateien in der Verzweigung, an die Sie übertragen (d. h. der Zielverzweigung).

Wenn Sie beispielsweise einen Commit an übertragen`main`, CodeCatalyst startet eine Workflow-Ausführung mit der `workflow`-Definitionsdatei und anderen Quelldateien auf `main`.

Wenn Sie beispielsweise einen Commit an übertragen`feature-branch-123`, CodeCatalyst startet eine Workflow-Ausführung mit der `Workflow`-Definitionsdatei und anderen Quelldateien auf `feature-branch-123`.

#### Triggers:

- Type: PUSH

#### Note

Wenn Sie möchten, dass eine Workflow-Ausführung nur gestartet wird, wenn Sie zu `pushenmain`, finden Sie weitere Informationen unter [Beispiel: Ein einfacher „Push to Main“-Auslöser](#).

### Beispiel: Ein einfacher „Push to Main“-Auslöser

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung startet, wenn Code an den `main` Zweig – und nur den `main` Zweig – in Ihrem Quell-Repository übertragen wird.



```
Triggers:  
- Type: PUSH  
  Branches:  
    - main
```

Beispiel: Ein einfacher Pull-Anforderungsauslöser

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung startet, wenn eine Pull-Anforderung in Ihrem Quell-Repository erstellt oder überarbeitet wird.

Wenn dieser Auslöser aktiviert ist, CodeCatalyst startet eine Workflow-Ausführung mit der Workflow-Definitionsdatei und anderen Quelldateien in der Verzweigung, aus der Sie abrufen (d. h. der Quellverzweigung).

Wenn Sie beispielsweise eine Pull-Anforderung mit einer Quellverzweigung namens `feature-123` und einer Zielverzweigung namens `erstellenmain`, CodeCatalyst startet eine Workflow-Ausführung mit der Workflow-Definitionsdatei und anderen Quelldateien auf `feature-123`.

```
Triggers:  
- Type: PULLREQUEST  
  Events:  
    - OPEN  
    - REVISION
```

Beispiel: Ein einfacher Zeitplanauslöser

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung jeden Montag bis Freitag um Mitternacht (UTC+0) startet.

Wenn dieser Auslöser aktiviert ist, CodeCatalyst startet eine einzelne Workflow-Ausführung für jede Verzweigung in Ihrem Quell-Repository, die eine Workflow-Definitionsdatei mit diesem Auslöser enthält.

Wenn Sie beispielsweise drei Verzweigungen in Ihrem Quell-Repository haben, `main`, `release-v1feature-123`, und jede dieser Verzweigungen enthält eine Workflow-Definitionsdatei mit dem folgenden Auslöser, CodeCatalyst startet drei Workflow-Ausführungen: eine mit den Dateien in `release-v1`, eine mit den Dateien in `main` und eine andere mit den Dateien in `feature-123`.

```
Triggers:  
- Type: SCHEDULE
```

```
Expression: "0 0 ? * MON-FRI *"
```

Weitere Beispiele für Cron-Ausdrücke, die Sie in der `-Expression`Eigenschaft verwenden können, finden Sie unter [Expression](#).

Beispiel: Ein Auslöser mit einem Zeitplan und Verzweigungen

Das folgende Beispiel zeigt einen Auslöser, der jeden Tag um 18:15 Uhr (UTC+0) eine Workflow-Ausführung startet.

Wenn dieser Auslöser aktiviert ist, CodeCatalyst startet eine Workflow-Ausführung mit den Dateien in der `main` Verzweigung und startet zusätzliche Ausführungen für jede Verzweigung, die mit `beginntrelease-`.

Wenn Sie beispielsweise Verzweigungen mit den Namen `main`, `release-v1bugfix-1`, und `bugfix-2` in Ihrem Quell-Repository haben, CodeCatalyst startet zwei Workflow-Ausführungen: eine mit den Dateien in `main` und eine andere mit den Dateien in `release-v1`. Workflow-Ausführungen für die `bugfix-1` Verzweigungen `bugfix-1` werden nicht gestartet.

Triggers:

- Type: SCHEDULE

Expression: "15 18 \* \* ? \*"

Branches:

- main

- release\-.\*

Weitere Beispiele für Cron-Ausdrücke, die Sie in der `-Expression`Eigenschaft verwenden können, finden Sie unter [Expression](#).

Beispiel: Ein Auslöser mit einem Zeitplan, einem Push und Verzweigungen

Das folgende Beispiel zeigt einen Auslöser, der jeden Tag eine Workflow-Ausführung um Mitternacht (UTC+0) startet und immer dann, wenn Code an die `main` Verzweigung übertragen wird.

In diesem Beispiel:

- Eine Workflow-Ausführung beginnt jeden Tag um Mitternacht. Die Workflow-Ausführung verwendet die Workflow-Definitionsdatei und andere Quelldateien in der `main` Verzweigung.
- Eine Workflow-Ausführung beginnt auch, wenn Sie einen Commit an die `main` Verzweigung übertragen. Die Workflow-Ausführung verwendet die Workflow-Definitionsdatei und andere Quelldateien in der Zielverzweigung (`main`).

**Triggers:**

- Type: SCHEDULE  
Expression: "0 0 \* \* ? \*"  
Branches:
  - main
- Type: PUSH  
Branches:
  - main

Weitere Beispiele für Cron-Ausdrücke, die Sie in der `-Expression`Eigenschaft verwenden können, finden Sie unter [Expression](#).

Beispiel: Ein Auslöser mit einem Pull und Verzweigungen

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung startet, wenn jemand eine Pull-Anforderung mit einem Zielzweig namens `öffnet` oder `ändert` `main`. Obwohl die in der `Triggers` Konfiguration angegebene Verzweigung ist `main`, verwendet die Workflow-Ausführung die Workflow-Definitionsdatei und andere Quelldateien in der Quellverzweigung (die Verzweigung, aus der Sie abrufen).

**Triggers:**

- Type: PULLREQUEST  
Branches:
  - main
- Events:
  - OPEN
  - REVISION

Beispiel: Ein Auslöser mit einem Pull-, Verzweig- und einem „CLOSED“-Ereignis

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung startet, wenn eine Pull-Anforderung für eine Verzweigung geschlossen wird, die mit `beginnt` `main`.

In diesem Beispiel:

- Wenn Sie eine Pull-Anforderung mit einem Zielzweig schließen, der mit `beginnt` `main`, beginnt eine Workflow-Ausführung automatisch mit der Workflow-Definitionsdatei und anderen Quelldateien im (jetzt geschlossenen) Quellzweig.
- Wenn Sie Ihr Quell-Repository so konfiguriert haben, dass Verzweigungen automatisch gelöscht werden, nachdem eine Pull-Anforderung zusammengeführt wurde, haben diese Verzweigungen

nie die Möglichkeit, in den `-CLOSED` Status überzugehen. Das bedeutet, dass zusammengeführte Verzweigungen den Pull-`CLOSED` Anforderungsauslöser nicht aktivieren. Die einzige Möglichkeit, den `CLOSED` Auslöser in diesem Szenario zu aktivieren, besteht darin, die Pull-Anforderung zu schließen, ohne sie zusammenzuführen.

**Triggers:**

- Type: `PULLREQUEST`
- Branches:
  - `main.*`
- Events:
  - `CLOSED`

Beispiel: Ein Auslöser mit Push, Verzweigungen und Dateien

Das folgende Beispiel zeigt einen Auslöser, der eine Workflow-Ausführung startet, wenn eine Änderung an der `filename.txt` Datei oder einer Datei im `src` Verzeichnis auf der `main` Verzweigung vorgenommen wird.

Wenn dieser Auslöser aktiviert ist, CodeCatalyst startet eine Workflow-Ausführung mit der Workflow-Definitionsdatei und anderen Quelldateien in der `main` Verzweigung.

**Triggers:**

- Type: `PUSH`
- Branches:
  - `main`
- FilesChanged:
  - `filename.txt`
  - `src\*.*`

## Arbeiten mit Variablen

Eine Variable ist ein Schlüssel-Wert-Paar, das Informationen enthält, auf die Sie in Ihrem CodeCatalyst Workflow verweisen können.

Es gibt zwei Arten von Variablen, die Sie in einem Workflow verwenden können:

- Benutzerdefinierte Variablen — Dies sind Schlüssel-Wert-Paare, die Sie definieren.
- Vordefinierte Variablen — Dies sind Schlüssel-Wert-Paare, die von einem Workflow automatisch ausgegeben werden. Sie müssen sie nicht definieren.

**Note**

CodeCatalyst unterstützt auch [GitHub Ausgabeparameter](#), die sich wie Variablen verhalten und in anderen Aktionen referenziert werden können. Weitere Informationen finden Sie unter [Mit GitHub Aktions-Ausgabeparametern arbeiten](#).

## Themen

- [Verwenden von benutzerdefinierten Variablen](#)
- [Verwenden vordefinierter Variablen](#)

## Verwenden von benutzerdefinierten Variablen

Benutzerdefinierte Variablen sind Schlüssel-Wert-Paare, die Sie definieren. Es gibt zwei Typen:

- Klartext-Variablen oder einfach Variablen — Dies sind Schlüssel-Wert-Paare, die Sie in der Workflow-Definitionsdatei im Klartext definieren.
- Secrets — Dies sind Schlüssel-Wert-Paare, die Sie auf einer separaten Secrets-Seite der CodeCatalyst Amazon-Konsole definieren. Der Schlüssel (Name) ist ein öffentliches Label, und der Wert enthält die Informationen, die Sie geheim halten möchten. Sie geben nur den Schlüssel in der Workflow-Definitionsdatei an. Verwenden Sie geheime Schlüssel anstelle von Kennwörtern und anderen vertraulichen Informationen in der Workflow-Definitionsdatei.

**Note**

Der Kürze halber wird in diesem Leitfaden der Begriff Variable für Klartext-Variable verwendet.

## Themen

- [Definition einer Variablen](#)
- [Ein Geheimnis definieren](#)
- [Eine Variable exportieren, sodass sie von anderen Aktionen verwendet werden kann](#)
- [In der Aktion, die sie definiert, auf eine Variable verweisen](#)
- [Verweisen auf eine Variable, die von einer anderen Aktion ausgegeben wurde](#)

- [Verweisen auf ein Geheimnis](#)
- [Beispiele](#)

## Definition einer Variablen

Sie können Variablen auf zwei Arten definieren:

- Im Inputs Abschnitt einer Workflow-Aktion — siehe [So definieren Sie eine Variable im Abschnitt „Eingaben“](#)
- Im Steps Abschnitt einer Workflow-Aktion — siehe [So definieren Sie eine Variable im Abschnitt „Schritte“](#)

### Note

Die Steps Methode funktioniert nur mit den Aktionen CodeCatalyst Build, Test und GitHub Actions, da dies die einzigen Aktionen sind, die einen Steps Abschnitt enthalten.

Beispiele finden Sie unter [Beispiele](#).

## Visual

Um eine Variable im Abschnitt „Eingaben“ zu definieren (visueller Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, für die Sie die Variable setzen möchten.
8. Wählen Sie Eingaben.
9. Wählen Sie unter Variablen — optional die Option Variable hinzufügen aus, und gehen Sie dann wie folgt vor:

Geben Sie eine Folge von Name/Wert-Paaren an, die die Eingabevariablen definieren, die Sie für die Aktion verfügbar machen möchten. Variablennamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Variablennamen zuzulassen.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um eine Variable im Abschnitt „Eingaben“ zu definieren (YAML-Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer Workflow-Aktion Code hinzu, der dem folgenden ähnelt:

```
action-name:
  Inputs:
    Variables:
      - Name: variable-name
        Value: variable-value
```

Weitere Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für Ihre Aktion.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Visual

Um eine Variable im Abschnitt „Schritte“ zu definieren (visueller Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion aus, für die Sie die Variable setzen möchten.
8. Wählen Sie Konfiguration.
9. Definieren Sie in Shell-Befehlen oder GitHubActions YAML, je nachdem, was verfügbar ist, eine Variable in den AktionenSteps, entweder explizit oder implizit.
  - Um die Variable explizit zu definieren, fügen Sie sie in einen Bash-Befehl direkt in den Abschnitt ein. Steps
  - Um eine Variable implizit zu definieren, geben Sie sie in einer Datei an, auf die im Abschnitt der Aktion verwiesen wird. Steps

Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für die Aktion.

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.



## YAML

Um eine Variable im Abschnitt „Schritte“ zu definieren (YAML-Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Definieren Sie in einer Workflow-Aktion eine Variable im Steps Abschnitt der Aktion, entweder explizit oder implizit.
  - Um die Variable explizit zu definieren, fügen Sie sie in einen Bash-Befehl direkt in den Steps Abschnitt ein.
  - Um eine Variable implizit zu definieren, geben Sie sie in einer Datei an, auf die im Abschnitt der Aktion verwiesen wird. Steps

Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für die Aktion.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

### Ein Geheimnis definieren

Sie definieren ein Geheimnis auf der Seite Secrets der CodeCatalyst Konsole. Weitere Informationen finden Sie unter [Arbeiten mit Secrets](#).

Sie könnten beispielsweise ein Geheimnis definieren, das wie folgt aussieht:

- Name (Schlüssel): **my-password**
- Wert: **^\*H3#!b9**

Nachdem das Geheimnis definiert wurde, können Sie den Schlüssel (**my-password**) des Geheimnisses in der Workflow-Definitionsdatei angeben. Ein Beispiel für diese Vorgehensweise finden Sie unter [Beispiel: Verweisen auf ein Geheimnis](#).

Eine Variable exportieren, sodass sie von anderen Aktionen verwendet werden kann

Verwenden Sie die folgenden Anweisungen, um eine Variable aus einer Aktion zu exportieren, sodass Sie in anderen Aktionen darauf verweisen können.

Bevor Sie eine Variable exportieren, beachten Sie Folgendes:

- Wenn Sie nur innerhalb der Aktion, in der sie definiert ist, auf die Variable verweisen müssen, müssen Sie sie nicht exportieren.
- Nicht alle Aktionen unterstützen den Export von Variablen. Um festzustellen, ob Ihre Aktion diese Funktion unterstützt, führen Sie die folgenden Anweisungen für den visuellen Editor durch und prüfen Sie, ob die Aktion auf der Registerkarte Ausgaben eine Schaltfläche „Variablen“ enthält. Falls ja, wird der Export von Variablen unterstützt.
- Informationen zum Exportieren einer Variablen aus einer GitHub Aktion finden Sie unter [Exportieren eines GitHub Ausgabeparameters, damit er von anderen Aktionen verwendet werden kann](#).

## Voraussetzung

Stellen Sie sicher, dass Sie die Variable definiert haben, die Sie exportieren möchten. Weitere Informationen finden Sie unter [Definition einer Variablen](#).

## Visual

Um eine Variable zu exportieren (visueller Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie Visual.

7. Wählen Sie im Workflow-Diagramm die Aktion aus, aus der Sie die Variable exportieren möchten.
8. Wählen Sie Ausgaben aus.
9. Wählen Sie unter Variablen — optional die Option Variable hinzufügen aus, und gehen Sie dann wie folgt vor:

Geben Sie den Namen einer Variablen an, die die Aktion exportieren soll. Diese Variable muss bereits im Steps Abschnitt Inputs oder derselben Aktion definiert sein.

10. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
11. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## YAML

Um eine Variable zu exportieren (YAML-Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in der Aktion, aus der Sie die Variable exportieren möchten, Code hinzu, der dem folgenden ähnelt:

```
action-name:  
  Outputs:  
    Variables:  
      - Name: variable-name
```

Weitere Beispiele finden Sie unter [Beispiele](#).

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

In der Aktion, die sie definiert, auf eine Variable verweisen

Verwenden Sie die folgenden Anweisungen, um in der Aktion, die sie definiert, auf eine Variable zu verweisen.

#### Note

Informationen zum Verweisen auf eine durch eine GitHub Aktion generierte Variable finden Sie unter [Referenzieren eines GitHub Ausgabeparameters](#).

#### Voraussetzung

Stellen Sie sicher, dass Sie die Variable definiert haben, auf die Sie verweisen möchten. Weitere Informationen finden Sie unter [Definition einer Variablen](#).

#### Visual

Nicht verfügbar. Wählen Sie YAML, um die YAML-Anweisungen anzuzeigen.

#### YAML

Um in der Aktion, die sie definiert, auf eine Variable zu verweisen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.

7. Fügen Sie in der CodeCatalyst Aktion, die die Variable definiert, auf die Sie verweisen möchten, die Variable mit der folgenden Bash-Syntax hinzu:

```
$variable-name
```

Beispiele:

```
MyAction:
  Configuration:
    Steps:
      - Run: $variable-name
```

Weitere Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie in den Referenzinformationen für Ihre Aktion im [Referenz zur Workflow-Definition](#).

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

Verweisen auf eine Variable, die von einer anderen Aktion ausgegeben wurde

Verwenden Sie die folgenden Anweisungen, um auf Variablen zu verweisen, die von anderen Aktionen ausgegeben wurden.

#### Note

Informationen zum Verweisen auf eine Variablenausgabe aus einer GitHub Aktion finden Sie unter [Referenzieren eines GitHub Ausgabeparameters](#).

Voraussetzung

Stellen Sie sicher, dass Sie die Variable exportiert haben, auf die Sie verweisen möchten. Weitere Informationen finden Sie unter [Eine Variable exportieren, sodass sie von anderen Aktionen verwendet werden kann](#).

Visual

Nicht verfügbar. Wählen Sie YAML, um die YAML-Anweisungen anzuzeigen.

## YAML

Um auf eine Variable zu verweisen, die von einer anderen Aktion ausgegeben wurde (YAML-Editor)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in der CodeCatalyst Aktion mithilfe der folgenden Syntax einen Verweis auf die Variable hinzu:

```
${action-group-name.action-name.variable-name}
```

Ersetze:

- *action-group-name* mit dem Namen der Aktionsgruppe, die die Aktion enthält, die eine Variable ausgibt.

### Note

Sie können weglassen, *action-group-name* wenn es keine Aktionsgruppe gibt oder wenn die Variable durch eine Aktion in derselben Aktionsgruppe erzeugt wird.

- *Aktionsname* mit dem Namen der Aktion, die die Variable ausgibt.
- *Variablenname mit dem Namen* der Variablen.

Beispiele:

```
MySecondAction:
  Configuration:
    Steps:
```

```
- Run: ${MyFirstAction.TIMESTAMP}
```

Weitere Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für Ihre Aktion.

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

## Verweisen auf ein Geheimnis

Anweisungen zum Verweisen auf ein Geheimnis in der Workflow-Definitionsdatei finden Sie unter [Verwenden eines Secrets](#)

Ein Beispiel finden Sie unter [Beispiel: Verweisen auf ein Geheimnis](#).

## Beispiele

Die folgenden Beispiele zeigen, wie Variablen in der Workflow-Definitionsdatei definiert und referenziert werden.

## Beispiele

- [Beispiel: Definition einer Variablen mithilfe der Eigenschaft Inputs](#)
- [Beispiel: Definieren einer Variablen mithilfe der Eigenschaft Steps](#)
- [Beispiel: Exportieren einer Variablen mithilfe der Outputs-Eigenschaft](#)
- [Beispiel: Verweisen auf eine Variable, die in derselben Aktion definiert wurde](#)
- [Beispiel: Verweisen auf eine Variable, die in einer anderen Aktion definiert wurde](#)
- [Beispiel: Verweisen auf ein Geheimnis](#)

## Beispiel: Definition einer Variablen mithilfe der Eigenschaft Inputs

Das folgende Beispiel zeigt Ihnen, wie Sie zwei Variablen definieren, VAR1 und VAR2, in einem Inputs Abschnitt.

```
Actions:  
  Build:
```

```
Identifizier: aws/build@v1
Inputs:
  Variables:
    - Name: VAR1
      Value: "My variable 1"
    - Name: VAR2
      Value: "My variable 2"
```

### Beispiel: Definieren einer Variablen mithilfe der Eigenschaft Steps

Das folgende Beispiel zeigt Ihnen, wie Sie eine DATE Variable in dem Steps Abschnitt explizit definieren.

```
Actions:
  Build:
    Identifizier: aws/build@v1
    Configuration:
      Steps:
        - Run: DATE=$(date +%m-%d-%y)
```

### Beispiel: Exportieren einer Variablen mithilfe der Outputs-Eigenschaft

Das folgende Beispiel zeigt Ihnen, wie Sie zwei Variablen definieren REPOSITORY-URI und TIMESTAMP diese mithilfe des Outputs Abschnitts exportieren.

```
Actions:
  Build:
    Identifizier: 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
```



## Beispiel: Verweisen auf eine Variable, die in derselben Aktion definiert wurde

Das folgende Beispiel zeigt Ihnen `MyBuildAction`, wie Sie eine `VAR1` Variable angeben und dann in derselben Aktion darauf verweisen, indem Sie `$VAR1`.

```
Actions:
  MyBuildAction:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: my-value
    Configuration:
      Steps:
        - Run: $VAR1
```

## Beispiel: Verweisen auf eine Variable, die in einer anderen Aktion definiert wurde

Das folgende Beispiel zeigt, wie Sie eine `TIMESTAMP` Variable in `BuildActionA` angeben, sie mithilfe der `Outputs` Eigenschaft exportieren und dann in `BuildActionB` using `${BuildActionA.TIMESTAMP}` darauf verweisen.

```
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.
```

## Beispiel: Verweisen auf ein Geheimnis

Das folgende Beispiel zeigt Ihnen, wie Sie auf ein my-password Geheimnis verweisen. Das my-password ist der Schlüssel des Geheimnisses. Der Schlüssel dieses Geheimnisses und der entsprechende Kennwortwert müssen auf der Seite Secrets der CodeCatalyst Konsole angegeben werden, bevor sie in der Workflow-Definitionsdatei verwendet werden können. Weitere Informationen finden Sie unter [Arbeiten mit Secrets](#).

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: curl -u LiJuan:${Secrets.my-password} https://example.com
```

## Verwenden vordefinierter Variablen

Vordefinierte Variablen sind Schlüssel-Wert-Paare, die von einem Workflow automatisch ausgegeben und Ihnen zur Verwendung in Workflow-Aktionen zur Verfügung gestellt werden.

Sie können vordefinierte Variablen in jeder Workflow-Aktion verwenden.

### Themen

- [Liste der vordefinierten Variablen](#)
- [Verweisen auf eine vordefinierte Variable](#)
- [Ermitteln Sie, welche vordefinierten Variablen Ihr Workflow ausgibt](#)
- [Beispiele](#)

### Liste der vordefinierten Variablen

In den folgenden Abschnitten finden Sie Informationen zu den vordefinierten Variablen, die automatisch von erzeugt werden CodeCatalyst.

#### Note

Diese Liste enthält nur vordefinierte Variablen, die von CodeCatalyst Workflows und [CodeCatalyst Aktionen ausgegeben werden](#). Wenn Sie andere Aktionstypen verwenden,

z. B. Aktionen oder GitHub CodeCatalyst Labs-Aktionen, finden Sie stattdessen weitere Informationen unter [Ermitteln Sie, welche vordefinierten Variablen Ihr Workflow ausgibt](#).

## Auflisten

- [Variablen BranchName "" und CommitId ""](#)
- [Aktionsvariablen erstellen und testen](#)
- [Aktionsvariablen „Amazon S3 Publish“](#)
- [AWS CDKAktionsvariablen „Bootstrap“](#)
- [AWS CDKAktionsvariablen „bereitstellen“](#)
- [AWS LambdaAktionsvariablen „aufrufen“](#)
- [Aktionsvariablen „AWS CloudFormationStack bereitstellen“](#)
- [Aktionsvariablen „In Amazon ECS bereitstellen“](#)
- [Aktionsvariablen „Auf Kubernetes-Cluster bereitstellen“](#)
- [Aktionsvariablen „Amazon ECS-Aufgabendefinition rendern“](#)
- [Aktionsvariablen „GitHub Aktionen“](#)

## Variablen BranchName "" und CommitId ""

CodeCatalyst erzeugt die Variablen BranchName "" und CommitId "", wenn Ihr Workflow ausgeführt wird. In der folgenden Tabelle finden Sie Informationen zu diesen Variablen.

| Schlüssel | Value (Wert)  |
|-----------|---|
| CommitId  | <p>Die Commit-ID, die den Status des Repositorys zum Zeitpunkt des Starts der Workflow-Ausführung darstellt.</p> <p>Beispiel: <code>example3819261db00a3ab59468c8b</code></p> <p>Weitere Informationen finden Sie auch unter: <a href="#">Beispiel: Verweisen auf die vordefinierte CommitId Variable ""</a>.</p> |

| Schlüssel  | Value (Wert)   |
|------------|--|
| BranchName | <p>Der Name des Branches, für den die Workflow-Ausführung gestartet wurde.</p> <p>Beispiele: <code>main</code>, <code>feature/branch</code> , <code>test-LiJuan</code> .</p> <p>Weitere Informationen finden Sie auch unter: <a href="#">Beispiel: Verweisen auf die vordefinierte BranchName Variable</a> "".</p> |

### Aktionsvariablen erstellen und testen

Die Build- und Testaktionen erzeugen nicht automatisch Variablen.

### Aktionsvariablen „Amazon S3 Publish“

Die Amazon S3 S3-Veröffentlichungsaktion erzeugt nicht automatisch Variablen.

### AWS CDKAktionsvariablen „Bootstrap“

Die AWS CDKBootstrap-Aktion erzeugt die folgenden vordefinierten Variablen.

| Schlüssel                | Value (Wert)   |
|--------------------------|--|
| Bereitstellungsplattform | <p>Der Name der Bereitstellungsplattform.</p> <p>Fest codiert auf. <code>AWS:CloudFormation</code></p>   |
| region                   | <p>Der Regionalcode desAWS-Region, für den der AWS CDK Bootstrap-Stack während der Workflow-Ausführung bereitgestellt wurde.</p> <p>Beispiel: <code>us-west-2</code></p> |
| Stack-ID                 | <p>Der Amazon-Ressourcenname (ARN) des bereitgestellten AWS CDK Bootstrap-Stacks.</p>  |

| Schlüssel                   | Value (Wert)  |
|-----------------------------|---|
|                             | <p>Beispiel: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-bootstrap-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code></p>   |
| BEREITSTELLUNG ÜBERSPRINGEN | <p>Der Wert von <code>true</code> gibt an, dass die Bereitstellung Ihres AWS CDK Bootstrap-Stacks während der Workflow-Ausführung übersprungen wurde. Eine Stack-Bereitstellung wird übersprungen, wenn sich der Stack seit der letzten Bereitstellung nicht geändert hat.</p> <p>Diese Variable wird nur erzeugt, wenn ihr Wert <code>true</code> ist.</p> <p>Fest codiert auf <code>true</code></p> |

### AWS CDK Aktionsvariablen „bereitstellen“

Die AWS CDK Bereitstellungsaktion erzeugt die folgenden Variablen.

| Schlüssel                | Value (Wert)  |
|--------------------------|---|
| Stack-ID                 | <p>Der Amazon-Ressourcenname (ARN) des AWS CDK Anwendungsstapels, für den während der Workflow-Ausführung bereitgestellt wurde.</p> <p>Beispiel: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-app-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code></p> |
| Bereitstellungsplattform | Der Name der Bereitstellungsplattform.  |

| Schlüssel                   | Value (Wert)  |
|-----------------------------|---|
| region                      | <p data-bbox="829 212 1414 243">Fest codiert auf. <code>AWS:CloudFormation</code></p> <p data-bbox="829 296 1463 422">Der Regionalcode vonAWS-Region, für den während der Workflow-Ausführung bereitgestellt wurde.</p> <p data-bbox="829 470 1133 501">Beispiel: <code>us-west-2</code></p>  |
| BEREITSTELLUNG ÜBERSPRINGEN | <p data-bbox="829 548 1479 821">Der Wert von <code>true</code> gibt an, dass die Bereitstellung Ihres AWS CDK Anwendungsstapels während der Workflow-Ausführung übersprungen wurde. Eine Stack-Bereitstellung wird übersprungen, wenn sich der Stack seit der letzten Bereitstellung nicht geändert hat.</p> <p data-bbox="829 869 1495 947">Diese Variable wird nur erzeugt, wenn ihr Wert <code>true</code> ist.</p> <p data-bbox="829 995 1146 1026">Fest codiert auf. <code>true</code></p> |

| Schlüssel                   | Value (Wert)   |
|-----------------------------|--|
| AWS CloudFormationVariablen | <p>Die Aktion „AWS CDKBereitstellen“ generiert nicht nur die zuvor aufgelisteten Variablen, sondern stellt auch CloudFormationAusgabevariablen als Workflow-Variablen zur Verwendung in nachfolgenden Workflow-Aktionen zur Verfügung. Standardmäßig macht die Aktion nur die ersten vier (oder weniger) CloudFormation Variablen verfügbar, die sie findet. Um festzustellen, welche verfügbar gemacht werden, führen Sie die AWS CDKBereitstellungsaktion einmal aus und schauen Sie dann auf der Seite mit den Ausführungsdetails auf der Registerkarte Variablen nach. Wenn die auf der Registerkarte Variablen aufgelisteten Variablen nicht Ihren Wünschen entsprechen, können Sie mithilfe der <code>CfnOutputVariables</code> YAML-Eigenschaft andere Variablen konfigurieren. Weitere Informationen finden Sie in der <a href="#">CfnOutputVariables</a> Eigenschaftsbeschreibung im <a href="#">Referenz zur Aktion „AWS CDK Bereitstellen“</a>.</p> |

### AWS LambdaAktionsvariablen „aufrufen“

Standardmäßig erzeugt die AWS LambdaAufrufaktion eine Variable pro Schlüssel der obersten Ebene in der Lambda-Antwortnutzlast.

Wenn die Antwortnutzlast beispielsweise wie folgt aussieht:

```
responsePayload = {
  "name": "Saanvi",
  "location": "Seattle",
  "department": {
    "company": "Amazon",
    "team": "AWS"
  }
}
```

```
}

```

... dann würde die Aktion die folgenden Variablen generieren.

| Schlüssel | Value (Wert)                        |
|-----------|-------------------------------------|
| Name      | Saanvi                              |
| location  | Seattle                             |
| Abteilung | {"Firma": „Amazon“, „Team“: "AWS,"} |

### Note

Sie können mithilfe der `ResponseFilters` YAML-Eigenschaft ändern, welche Variablen generiert werden. Weitere Informationen finden Sie unter [ResponseFilters](#) im „[AWS Lambda Aufruf](#)“-Aktionsreferenz.

Aktionsvariablen „AWS CloudFormationStack bereitstellen“

Die Aktion `AWS CloudFormationStack bereitstellen` erzeugt die folgenden Variablen.

| Schlüssel                | Value (Wert)   |
|--------------------------|--|
| Bereitstellungsplattform | Der Name der Bereitstellungsplattform.<br><br>Fest codiert auf. <code>AWS:CloudFormation</code>  |
| region                   | Der Regionalcode von AWS-Region, für den während der Workflow-Ausführung bereitgestellt wurde.<br><br>Beispiel: <code>us-west-2</code> |
| Stack-ID                 | Der Amazon-Ressourcenname (ARN) des bereitgestellten Stacks.   |



| Schlüssel | Value (Wert)   |
|-----------|--|
|           | Beispiel: <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cfn-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code> |

### Aktionsvariablen „In Amazon ECS bereitstellen“

Die Aktion Deploy to Amazon ECS erzeugt die folgenden Variablen.

| Schlüssel                | Value (Wert)  |
|--------------------------|---|
| Cluster                  | Der Name des Amazon ECS-Clusters, auf dem während der Workflow-Ausführung bereitgestellt wurde.<br><br>Beispiel: <code>codecatalyst-ecs-cluster</code>  |
| Bereitstellungsplattform | Der Name der Bereitstellungsplattform.<br><br>Fest codiert auf. <code>AWS:ECS</code>  |
| Service nicht zulässig   | Der Name des Amazon ECS-Service, für den während der Workflow-Ausführung bereitgestellt wurde.<br><br>Beispiel: <code>codecatalyst-ecs-service</code>   |
| task-definition-arn      | Der Amazon-Ressourcenname (ARN) der Aufgabendefinition, die während der Workflow-Ausführung registriert wurde.<br><br>Beispiel: <code>arn:aws:ecs:us-west-2:111122223333:task-definition/codecatalyst-task-def:8</code><br><br>Der Wert <code>:8</code> im vorherigen Beispiel gibt die Revision an, die registriert wurde. |

| Schlüssel           | Value (Wert)   |
|---------------------|--|
| Bereitstellungs-URL | <p>Ein Link zur Registerkarte Ereignisse der Amazon ECS-Konsole, auf der Sie Details zur Amazon ECS-Bereitstellung im Zusammenhang mit der Workflow-Ausführung einsehen können.</p> <p>Beispiel: <code>https://console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/codecatalyst-ecs-cluster/services/codecatalyst-ecs-service/events</code></p> |
| region              | <p>Der Regionalcode von AWS-Region, für den während der Workflow-Ausführung bereitgestellt wurde.</p> <p>Beispiel: <code>us-west-2</code></p>  |

### Aktionsvariablen „Auf Kubernetes-Cluster bereitstellen“

Die Cluster-Aktion „Auf Kubernetes bereitstellen“ erzeugt die folgenden Variablen.

| Schlüssel                | Value (Wert)  |
|--------------------------|---|
| Cluster                  | <p>Der Amazon.com-Ressourcenname (ARN) des Amazon EKS-Clusters, auf dem während der Workflow-Ausführung bereitgestellt wurde.</p> <p>Beispiel: <code>arn:aws:eks:us-west-2:111122223333:cluster/codecatalyst-eks-cluster</code></p> |
| Bereitstellungsplattform | <p>Der Name der Bereitstellungsplattform.</p> <p>Fest codiert auf. <code>AWS:EKS</code></p>   |

| Schlüssel  | Value (Wert)   |
|------------|--|
| Metadaten  | Reserved Instances. Metadaten im JSON-Format, die sich auf den Cluster beziehen, der während der Workflow-Ausführung bereitgestellt wurde.   |
| Namespace  | Der Kubernetes-Namespace, in dem der Cluster bereitgestellt wurde.<br><br>Beispiel: default  |
| Ressourcen | Reserved Instances. Metadaten im JSON-Format, die sich auf die Ressourcen beziehen, die während der Workflow-Ausführung bereitgestellt wurden.   |
| server     | Der Name des API-Serverendpunkts, den Sie für die Kommunikation mit Ihrem Cluster mithilfe von Verwaltungstools wie verwenden können. kubectl<br><br>Weitere Informationen zum API-Serviceendpunkt finden Sie unter <a href="#">Amazon EKS-Cluster-Endpunktzugriffskontrolle</a> im Amazon EKS-Benutzerhandbuch.<br><br>Beispiel: <code>https://random-st<br/>ring .gr7.us-west-2.eks.amazonaws.com</code> |

### Aktionsvariablen „Amazon ECS-Aufgabendefinition rendern“

Die Aktion „Amazon ECS-Aufgabendefinition rendern“ erzeugt die folgenden Variablen.

| Schlüssel          | Value (Wert)  |
|--------------------|---|
| Aufgabendefinition | <p>Der Name der Aufgabendefinitionsdatei, die durch die Amazon ECS-Aufgabendefinitionsaktion Rendern aktualisiert wurde. Der Name hat folgendes Format: <code>task-definition-<i>random-string</i>.json</code>.</p> <p>Beispiel: <code>task-definition--259-0a2r7gx1TF5Xr.json</code></p> |

## Aktionsvariablen „GitHub Aktionen“

Die Aktion "GitHub Aktionen" erzeugt nicht automatisch Variablen.

### Verweisen auf eine vordefinierte Variable

Verwenden Sie die folgenden Anweisungen, um auf eine vordefinierte Variable zu verweisen.

### Voraussetzung

Ermitteln Sie den Namen der vordefinierten Variablen, auf die Sie verweisen möchten, z. `CommitId`. B. Weitere Informationen finden Sie unter [Ermitteln Sie, welche vordefinierten Variablen Ihr Workflow ausgibt](#).

### Visual

Nicht verfügbar. Wählen Sie YAML, um die YAML-Anweisungen anzuzeigen.

### YAML

Um auf eine vordefinierte Variable zu verweisen (YAML-Editor)


1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.

5. Wählen Sie Edit (Bearbeiten) aus.
6. Wählen Sie YAML.
7. Fügen Sie in einer CodeCatalyst Aktion die vordefinierte Variablenreferenz mit der folgenden Syntax hinzu:

```
${action-group-name.action-name-or-WorkflowSource.variable-name}
```

Ersetzen Sie:

- *action-group-name* mit dem Namen der Aktionsgruppe.

 Note

Sie können weglassen, *action-group-name* wenn es keine Aktionsgruppe gibt oder wenn die Variable durch eine Aktion in derselben Aktionsgruppe erzeugt wird.

- *action-name-or- WorkflowSource* mit:

Der Name der Aktion, die die Variable ausgibt.

oder

WorkflowSource, wenn die Variable die CommitId Variable BranchName oder ist.

- *Variablenname* mit dem Namen der Variablen.

Beispiele:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${MyFirstECSAction.cluster}
```

Ein weiteres Beispiel:

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${WorkflowSource.CommitId}
```

Weitere Beispiele finden Sie unter [Beispiele](#). Weitere Informationen finden Sie unter [Referenz zur Workflow-Definition](#) Für Ihre Aktion.

- (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
- Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.

Ermitteln Sie, welche vordefinierten Variablen Ihr Workflow ausgibt

Sie können auf zwei Arten bestimmen, welche vordefinierten Variablen Ihr Workflow ausgibt:

- Führen Sie den Workflow einmal aus. Nach Abschluss des Laufs werden die vom Workflow ausgegebenen Variablen auf der Registerkarte Variablen der Seite mit den Ausführungsdetails angezeigt. Weitere Informationen finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).
- Konsultieren Sie die [Liste der vordefinierten Variablen](#). In dieser Referenz sind der Variablenname (Schlüssel) und der Wert für jede vordefinierte Variable aufgeführt.

#### Note

Die maximale Gesamtgröße der Variablen eines Workflows ist unter [aufgeführt](#)[Kontingente für Workflows in CodeCatalyst](#). Wenn die Gesamtgröße das Maximum überschreitet, schlägt die Aktion, die nach dem Erreichen des Maximums ausgeführt wird, möglicherweise fehl.

## Beispiele

Die folgenden Beispiele zeigen, wie Sie in der Workflow-Definitionsdatei auf vordefinierte Variablen verweisen.

### Beispiele

- [Beispiel: Verweisen auf die vordefinierte CommitId Variable ""](#)
- [Beispiel: Verweisen auf die vordefinierte BranchName Variable ""](#)

### Beispiel: Verweisen auf die vordefinierte CommitId Variable ""

Das folgende Beispiel zeigt Ihnen, wie Sie in der MyBuildAction Aktion auf die CommitId vordefinierte Variable verweisen. Die CommitId Variable wird automatisch von ausgegeben CodeCatalyst.

Obwohl das Beispiel zeigt, dass die Variable in der Build-Aktion verwendet wird, können Sie sie CommitId in jeder beliebigen Aktion verwenden.

```
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}
```

### Beispiel: Verweisen auf die vordefinierte BranchName Variable ""

Das folgende Beispiel zeigt Ihnen, wie Sie in der CDKDeploy Aktion auf die BranchName vordefinierte Variable verweisen. Die BranchName Variable wird automatisch von ausgegeben CodeCatalyst.

Obwohl das Beispiel zeigt, dass die Variable in der AWS CDKBereitstellungsaktion verwendet wird, können Sie sie BranchName in jeder beliebigen Aktion verwenden.

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    StackName: app-stack-${WorkflowSource.BranchName}
```

## Referenz zur Workflow-Definition

Im Folgenden finden Sie die Referenzdokumentation für die Workflow-Definitionsdatei.

Eine Workflow-Definitionsdatei ist eine YAML-Datei, die Ihren Workflow beschreibt. Die Datei wird in einem `~/ .codecatalyst/workflows/` Ordner im Stammverzeichnis Ihres [Quell-Repositorys](#) gespeichert. Die Datei kann die Erweiterung `.yml` oder `.yaml` haben.

Um die Workflow-Definitionsdatei zu erstellen und zu bearbeiten, können Sie einen Editor wie vim oder den visuellen Editor oder den YAML-Editor der CodeCatalyst Konsole verwenden. Weitere Informationen finden Sie unter [Verwenden der visuellen Editoren und der YAML-Editoren der CodeCatalyst Konsole](#).

#### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende Benutzeroberflächenelemente im visuellen Editor. Verwenden Sie Strg+F, um nach einem UI-Element zu suchen. Das Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

## Themen

- [Beispiel für eine Workflow-Definitionsdatei](#)
- [Richtlinien und Konventionen zur Syntax](#)
- [Immobilien auf höchstem Niveau](#)
- [Aktionsreferenz zum Erstellen und Testen](#)
- [Referenz zur Aktion „Amazon S3 veröffentlichen“](#)
- [Referenz zur Aktion „AWS CDK bootstrap“](#)
- [Referenz zur Aktion „AWS CDK Bereitstellen“](#)
- [„AWS Lambda Aufruf“-Aktionsreferenz](#)
- [Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“](#)
- [Referenz zur Aktion „Deploy to Amazon ECS“](#)
- [Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#)
- [Referenz zur Aktion „GitHub Aktionen“](#)
- [Referenz zur Aktion „Amazon-ECS-Aufgabendefinition rendern“](#)

## Beispiel für eine Workflow-Definitionsdatei

Im Folgenden finden Sie ein Beispiel für eine einfache Workflow-Definitionsdatei. Sie umfasst einige Eigenschaften der obersten Ebene, einen Triggers Abschnitt und einen Actions Abschnitt mit



zwei Aktionen: Build und Test. Weitere Informationen finden Sie unter [Informationen zur Workflow-Definitionsdatei](#).

```
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
```

## Richtlinien und Konventionen zur Syntax

In diesem Abschnitt werden die Syntaxregeln für die Workflow-Definitionsdatei sowie die in dieser Referenzdokumentation verwendeten Namenskonventionen beschrieben.

### Richtlinien für die YAML-Syntax

Die Workflow-Definitionsdatei ist in YAML geschrieben und folgt der [YAML 1.1-Spezifikation, sodass alles, was in dieser Spezifikation](#) zulässig ist, auch in der Workflow-YAML zulässig ist. Wenn Sie mit YAML noch nicht vertraut sind, finden Sie hier einige kurze Richtlinien, um sicherzustellen, dass Sie gültigen YAML-Code angeben.

- Berücksichtigung von Groß- und Kleinschreibung: In der Workflow-Definitionsdatei wird zwischen Groß- und Kleinschreibung unterschieden. Stellen Sie daher sicher, dass Sie die in dieser Dokumentation beschriebene Groß- und Kleinschreibung verwenden.
- Sonderzeichen: Wir empfehlen die Verwendung von Anführungszeichen oder doppelten Anführungszeichen für Eigenschaftswerte, die eines der folgenden Sonderzeichen enthalten: { } [, ], ,, , & \* # , ? , | , < - , > , , , , = , ! % , und @ : ` ,

Wenn Sie die Anführungszeichen nicht angeben, können die oben aufgeführten Sonderzeichen unerwartet interpretiert werden.

- Eigenschaftsnamen: Eigenschaftsnamen (im Gegensatz zu Eigenschaftswerten) sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen oder doppelte Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Eigenschaftsnamen zuzulassen.

Nicht zulässig:

```
'My#Build@action'
```

```
My#Build@action
```

```
My Build Action
```

Zulässig:

```
My-Build-Action_1
```

- Escape-Codes: Wenn Ihr Immobilienwert Escape-Codes enthält (z. B. `\n` oder `\t`), befolgen Sie diese Richtlinien:
  - Verwenden Sie einfache Anführungszeichen, um den Escape-Code als Zeichenfolge zurückzugeben. Gibt beispielsweise `'my string \n my string'` die Zeichenfolge zurück `my string \n my string`.
  - Verwenden Sie doppelte Anführungszeichen, um den Escape-Code zu analysieren. Gibt zum `"my string \n my new line"` Beispiel zurück:

```
my string
my new line
```

- Kommentare: Kommentare im Vorwort mit #.

## Beispiel:

```
Name: MyWorkflow
# This is a comment.
SchemaVersion: 1.0
```

- Dreifacher Gedankenstrich (---): Nicht --- in Ihrem YAML-Code verwenden. CodeCatalyst ignoriert alles nach dem. ---

## Namenskonventionen

In diesem Handbuch beziehen wir uns mit den Begriffen Eigenschaft und Abschnitt auf die wichtigsten Elemente in einer Workflow-Definitionsdatei.

- Eine Eigenschaft ist jedes Element, das einen Doppelpunkt (:) enthält. Im folgenden Codeausschnitt sind beispielsweise alle folgenden Eigenschaften: `Name`, `SchemaVersion`, `RunMode`, `TriggersType`, und `Branches`
- Ein Abschnitt ist eine Eigenschaft, die Untereigenschaften hat. Im folgenden Codeausschnitt gibt es einen Abschnitt `Triggers`

### Note

In diesem Handbuch werden „Abschnitte“ manchmal als „Eigenschaften“ bezeichnet und umgekehrt, je nach Kontext.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

## Immobilien auf höchstem Niveau

Im Folgenden finden Sie die Referenzdokumentation für die Eigenschaften der obersten Ebene in der Workflow-Definitionsdatei.

```
# Name
Name: workflow-name

# Schema version
SchemaVersion: 1.0

# Run mode
RunMode: QUEUED|SUPERSEDED|PARALLEL

# Compute
Compute:
...

# Triggers
Triggers:
...

# Actions
Actions:
...
```

### Name

(Erforderlich)

Der Name des Workflows. Der Workflow-Name wird in der Workflow-Liste angezeigt und in Benachrichtigungen und Protokollen erwähnt. Der Workflowname und der Name der Workflow-Definitionsdatei können übereinstimmen, oder Sie können sie unterschiedlich benennen. Workflow-Namen müssen nicht eindeutig sein. Workflow-Namen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Workflow-Namen zuzulassen.

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Eigenschaften/Workflow-Name

## SchemaVersion

(Erforderlich)

Die Schemaversion der Workflow-Definition. Der einzige gültige Wert ist derzeit 1.0.

Entsprechende Benutzeroberfläche: keine

## RunMode

(Optional)

Wie CodeCatalyst geht man mit mehreren Durchläufen um? Sie können einen der folgenden Werte verwenden:

- **QUEUED**— Mehrere Läufe werden in die Warteschlange gestellt und nacheinander ausgeführt. Sie können bis zu 50 Läufe in einer Warteschlange haben.
- **SUPERSEDED**— Mehrere Läufe werden in die Warteschlange gestellt und nacheinander ausgeführt. Eine Warteschlange kann nur einen Lauf haben. Wenn also zwei Läufe zusammen in derselben Warteschlange landen, ersetzt der spätere Lauf den früheren Lauf (übernimmt ihn), und der frühere Lauf wird abgebrochen.
- **PARALLEL**— Es finden mehrere Läufe gleichzeitig statt.

Wenn diese Eigenschaft weggelassen wird, ist die Standardeinstellung **QUEUED**.

Weitere Informationen finden Sie unter [Konfiguration von Läufen in Warteschlangen, abgelösten und parallel Läufen](#).

Entsprechende Benutzeroberfläche: Visual Editor/Workflow-Eigenschaften/Erweitert/Ausführungsmodus

## Compute

(Optional)

Die Rechen-Engine, die zur Ausführung Ihrer Workflow-Aktionen verwendet wird. Sie können die Berechnung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht beide. Wenn auf Workflow-Ebene angegeben, gilt die Rechenkonfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instanz ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Weitere Informationen zu Compute finden Sie unter [Arbeiten mit Docker-Images für Rechen- und Laufzeitumgebungen](#).

Entsprechende Benutzeroberfläche: keine

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
  SharedInstance: true | false
```

## Type

(Compute/Type)

(Erforderlich, Compute wenn gesetzt)

Der Typ der Compute Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsläufen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Visual Editor/Workflow-Eigenschaften/Erweitert/Berechnungstyp

## Fleet

(Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden sollen. Bei bedarfsgesteuerten Flotten stellt der Workflow zu Beginn einer Aktion die

benötigten Ressourcen bereit, und die Maschinen werden zerstört, wenn die Aktion abgeschlossen ist. Beispiele für Flotten auf Abruf: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Weitere Informationen zu Flotten auf Abruf finden Sie unter [Flotteneigenschaften auf Abruf](#)

Bei bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen, um Ihre Workflow-Aktionen auszuführen. Diese Maschinen bleiben inaktiv und können sofort Aktionen ausführen. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#)

Wenn Fleet es weggelassen wird, ist die Standardeinstellung `Linux.x86-64.Large`

Weitere Informationen zu Rechenflotten finden Sie unter [Über Rechenflotten](#).

Entsprechende Benutzeroberfläche: Visual Editor/Workflow-Eigenschaften/Erweitert/Compute Fleet

SharedInstance

(Compute/SharedInstance)

(Optional)

Geben Sie die Funktion zur gemeinsamen Nutzung von Rechenleistung für Ihre Aktionen an. Bei Compute Sharing werden Aktionen in einem Workflow auf derselben Instanz ausgeführt (Image der Laufzeitumgebung). Sie können einen der folgenden Werte verwenden:

- `TRUE` bedeutet, dass das Laufzeitumgebungs-Image von allen Workflow-Aktionen gemeinsam genutzt wird.
- `FALSE` bedeutet, dass für jede Aktion in einem Workflow ein separates Laufzeitumgebungs-Image gestartet und verwendet wird, sodass Sie Ressourcen wie Artefakte und Variablen nicht ohne zusätzliche Konfiguration gemeinsam nutzen können.

Weitere Informationen zur gemeinsamen Nutzung von Compute finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

Triggers

(Optional)

Eine Sequenz von einem oder mehreren Triggern für diesen Workflow. Wenn kein Trigger angegeben ist, müssen Sie Ihren Workflow manuell starten.

Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Diagramm/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)

(Erforderlich, wenn gesetzt) Triggers

Geben Sie den Triggertyp an. Sie können einen der folgenden Werte verwenden:

- Push (visueller Editor) oder PUSH (YAML-Editor)



Ein Push-Trigger startet einen Workflow-Lauf, wenn eine Änderung in Ihr Quell-Repository übertragen wird. Bei der Workflow-Ausführung werden die Dateien in dem Branch verwendet, in den Sie pushen (das ist der Ziel-Branch).

- Pull-Request (visueller Editor) oder PULLREQUEST (YAML-Editor)

Ein Pull-Request-Trigger startet einen Workflow-Lauf, wenn ein Pull-Request in Ihrem Quell-Repository geöffnet, aktualisiert oder geschlossen wird. Bei der Workflow-Ausführung werden die Dateien in dem Branch verwendet, aus dem Sie abrufen (das ist der Quell-Branch).

- Zeitplan (visueller Editor) oder SCHEDULE (YAML-Editor)

Ein Zeitplan-Trigger startet Workflow-Läufe nach einem Zeitplan, der durch einen von Ihnen angegebenen Cron-Ausdruck definiert ist. Für jeden Branch in Ihrem Quell-Repository wird ein separater Workflow-Lauf gestartet, der die Dateien des Branches verwendet. (Verwenden Sie das Feld Branches (visueller Editor) oder die Branches Eigenschaft (YAML-Editor), um die Branches einzuschränken, für die der Trigger aktiviert wird.)

Beachten Sie bei der Konfiguration eines Zeitplan-Triggers die folgenden Richtlinien:

- Verwenden Sie nur einen Zeitplan-Trigger pro Workflow.
- Wenn Sie in Ihrem CodeCatalyst Bereich mehrere Workflows definiert haben, empfehlen wir, nicht mehr als 10 davon so zu planen, dass sie gleichzeitig starten.
- Stellen Sie sicher, dass Sie den Cron-Ausdruck des Triggers so konfigurieren, dass genügend Zeit zwischen den Läufen liegt. Weitere Informationen finden Sie unter [Expression](#).

Beispiele finden Sie unter [Beispiele für Auslöser](#).

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Diagramm/Trigger/Triggertyp

## Events

(Triggers/Events)

(Erforderlich, wenn der Trigger auf gesetzt ist) Type PULLREQUEST

Geben Sie den Typ der Pull-Request-Ereignisse an, mit denen eine Workflow-Ausführung gestartet wird. Die folgenden Werte sind gültig:

- Eine Pull-Anfrage wird erstellt (visueller Editor) oder OPEN (YAML-Editor)

Der Workflow-Lauf wird gestartet, wenn ein Pull-Request erstellt wird.

- Die Pull-Anfrage ist geschlossen (visueller Editor) oder CLOSED (YAML-Editor)

Der Workflow-Lauf wird gestartet, wenn ein Pull-Request geschlossen wird. Das Verhalten des CLOSED Ereignisses ist knifflig und lässt sich am besten anhand eines Beispiels verstehen.

Weitere Informationen finden Sie unter [Beispiel: Ein Auslöser mit einem Pull-, Verzweig- und einem „CLOSED“-Ereignis](#).

- Eine neue Version wird für den Pull-Request (visueller Editor) oder **REVISION** (YAML-Editor) erstellt

Der Workflow-Lauf wird gestartet, wenn eine Revision eines Pull-Requests erstellt wird. Die erste Revision wird erstellt, wenn der Pull-Request erstellt wird. Danach wird jedes Mal, wenn jemand einen neuen Commit an den im Pull Request angegebenen Quell-Branch pusht, eine neue Revision erstellt. Wenn Sie das REVISION Ereignis in Ihren Pull-Request-Trigger aufnehmen, können Sie das OPEN Ereignis weglassen, da es eine Obermenge von REVISION ist. OPEN

Sie können mehrere Ereignisse in demselben Pull-Request-Trigger angeben.

Beispiele finden Sie unter [Beispiele für Auslöser](#).

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Diagramm/Trigger/Ereignisse für Pull-Requests

Branches

(Triggers/Branches)

(Optional)

Geben Sie die Branches in Ihrem Quell-Repository an, die der Trigger überwacht, um zu wissen, wann ein Workflow-Lauf gestartet werden muss. Sie können Regex-Muster verwenden, um Ihre Branch-Namen zu definieren. Verwenden Sie dies beispielsweise, um alle Zweige `main.*` abzugleichen, die mit `main` beginnen.

Die anzugebenden Zweige unterscheiden sich je nach Triggertyp:

- Geben Sie für einen Push-Trigger die Zweige an, auf die Sie pushen möchten, d. h. die Zielzweige. Pro übereinstimmender Verzweigung wird ein Workflow-Lauf gestartet, wobei die Dateien im entsprechenden Zweig verwendet werden.

Beispiele: `main.*`, `mainline`

- Geben Sie für einen Pull-Request-Trigger die Branches an, in die Sie pushen möchten, also die Ziel-Banches. Pro zugeordnetem Branch wird ein Workflow-Lauf gestartet, wobei die Workflow-Definitionsdatei und die Quelldateien im Quellzweig (nicht im passenden Branch) verwendet werden.

Beispiele: `main.*`, `mainline`, `v1\-.*` (entspricht Verzweigungen, die mit `beginnev1-` beginnen)

- Geben Sie für einen Zeitplan-Trigger die Zweige an, die die Dateien enthalten, die bei Ihrem geplanten Lauf verwendet werden sollen. Pro zugeordnetem Zweig wird ein Workflow-Lauf gestartet, wobei die Workflow-Definitionsdatei und die Quelldateien im entsprechenden Zweig verwendet werden.

Beispiele: `main.*`, `version\-1\.*`

#### Note

Wenn Sie keine Verzweigungen angeben, überwacht der Trigger alle Verzweigungen in Ihrem Quell-Repository und startet eine Workflow-Ausführung mit der Workflow-Definitionsdatei und den Quelldateien in:

- Der Branch, auf den Sie pushen (für Push-Trigger). Weitere Informationen finden Sie unter [Beispiel: Ein einfacher Code-Push-Auslöser](#).
- Der Branch, aus dem Sie abrufen (für Pull-Request-Trigger). Weitere Informationen finden Sie unter [Beispiel: Ein einfacher Pull-Anforderungsauslöser](#).
- Alle Branches (für Zeitplan-Trigger). Pro Branch in Ihrem Quell-Repository wird ein Workflow-Lauf gestartet. Weitere Informationen finden Sie unter [Beispiel: Ein einfacher Zeitplanauslöser](#).

Weitere Informationen zu Branches und Triggern finden Sie unter [Überlegungen zu Auslösern bei der Verzweigung](#).

Weitere Beispiele finden Sie unter [Beispiele für Auslöser](#).

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Diagramm/Trigger/Zweige

## FilesChanged

(Triggers/FilesChanged)

(Optional, wenn der Trigger auf, oder gesetzt ist. Type PUSH PULLREQUEST Wird nicht unterstützt, wenn der Trigger auf gesetzt Type ist SCHEDULE.)

Geben Sie die Dateien oder Ordner in Ihrem Quell-Repository an, die der Trigger überwacht, um zu wissen, wann eine Workflow-Ausführung gestartet werden muss. Sie können reguläre Ausdrücke verwenden, um Dateinamen oder Pfade abzugleichen.

Beispiele finden Sie unter [Beispiele für Auslöser](#).

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Diagramm/Trigger/geänderte Dateien

## Expression

(Triggers/Expression)

(Erforderlich, wenn der Trigger auf gesetzt ist) Type SCHEDULE

Geben Sie den Cron-Ausdruck an, der beschreibt, wann Ihre geplanten Workflow-Ausführungen ausgeführt werden sollen.

In Cron-Ausdrücken wird die folgende Syntax mit sechs Feldern CodeCatalyst verwendet, wobei jedes Feld durch ein Leerzeichen getrennt ist:

*Minuten , Stunden days-of-month, Monat, Jahr days-of-week*

Beispiele für Cron-Ausdrücke

| Minuten | Stunden | Tage des Monats | Monat | Wochentage | Jahr | Bedeutung  |
|---------|---------|-----------------|-------|------------|------|--|
| 0       | 0       | ?               | *     | MO-FR      | *    | Führt jeden Montag bis Freitag um Mitternacht (UTC +0) einen Workflow aus. |

| Minuten | Stunden | Tage des Monats | Monat | Wochentage | Jahr | Bedeutung   |
|---------|---------|-----------------|-------|------------|------|---|
| 0       | 2       | *               | *     | ?          | *    | Führt jeden Tag um 2:00 Uhr (UTC+0) einen Workflow aus.   |
| 15      | 22      | *               | *     | ?          | *    | Führt jeden Tag um 22:15 Uhr (UTC +0) einen Workflow aus.   |
| 0/30    | 22-2    | ?               | *     | SAT-SUN    | *    | Führt samstags bis sonntags alle 30 Minuten einen Workflow zwischen 22:00 Uhr am Starttag und 2:00 Uhr am Folgetag (UTC+0) aus. |

| Minuten | Stunden | Tage des Monats | Monat | Wochentage | Jahr      | Bedeutung   |
|---------|---------|-----------------|-------|------------|-----------|---|
| 45      | 13      | L               | *     | ?          | 2023-2027 | Führt am letzten Tag des Monats zwischen 2023 und einschließlich 2027 um 13:45 Uhr (UTC +0) einen Workflow aus. |

Achten Sie bei der Angabe von Cron-Ausdrücken in darauf CodeCatalyst, dass Sie die folgenden Richtlinien beachten:

- Geben Sie einen einzelnen Cron-Ausdruck pro SCHEDULE Trigger an.
- Schließen Sie den Cron-Ausdruck im YAML-Editor in doppelte Anführungszeichen (") ein.
- Geben Sie die Uhrzeit in koordinierter Weltzeit (UTC) an. Andere Zeitzonen werden nicht unterstützt.
- Konfigurieren Sie mindestens 30 Minuten zwischen den Läufen. Eine schnellere Trittfrequenz wird nicht unterstützt.
- Geben Sie das *days-of-week* Feld *days-of-month* oder an, aber nicht beide. Wenn Sie in einem der Felder einen Wert oder ein Sternchen (\*) angeben, müssen Sie in dem anderen Feld ein Fragezeichen (?) verwenden. Das Sternchen bedeutet „alle“ und das Fragezeichen bedeutet „beliebig“.

Weitere Beispiele für Cron-Ausdrücke und Informationen zu Platzhaltern wie ? \*L, und finden Sie in der [Referenz zu Cron-Ausdrücken](#) im EventBridge Amazon-Benutzerhandbuch. Cron-Ausdrücke in EventBridge und CodeCatalyst funktionieren genauso.

Beispiele für Zeitplan-Trigger finden Sie unter [Beispiele für Auslöser](#).

Entsprechende Benutzeroberfläche: visueller Editor/Workflow-Diagramm/Trigger/Zeitplan

## Aktionen

Eine Abfolge von einer oder mehreren Aktionen für diesen Workflow. CodeCatalyst unterstützt mehrere Aktionstypen, z. B. Build- und Testaktionen, die unterschiedliche Funktionen bieten. Jeder Aktionstyp hat:

- eine `Identifier` Eigenschaft, die die eindeutige, hartcodierte ID der Aktion angibt. `aws/build@v1` identifiziert beispielsweise die Build-Aktion.
- ein `Configuration` Abschnitt, der Eigenschaften enthält, die für die Aktion spezifisch sind.

Weitere Informationen zu den einzelnen Aktionstypen finden Sie in der zugehörigen Referenzdokumentation unter [Referenz zur Workflow-Definition](#).

Im Folgenden finden Sie die YAML-Referenz für Aktionen und Aktionsgruppen in der Workflow-Definitionsdatei.

Weitere Informationen zu Aktionen finden Sie unter [Verwenden von Aktionen](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Actions:
  action-name:
    Identifier: action-identifier
    Configuration:
      ...
  #Action groups
  action-group-name:
    Actions:
      ...
```

### Aktionsname

*(Aktionsname) Actions/*

(Erforderlich)

Ersetzen Sie *Aktionsname* durch einen Namen, den Sie der Aktion geben möchten. Aktionsnamen müssen innerhalb des Workflows eindeutig sein und dürfen nur alphanumerische Zeichen, Bindestriche und Unterstriche enthalten. Weitere Informationen zu Syntaxregeln finden Sie unter [Richtlinien für die YAML-Syntax](#)

Weitere Informationen zu Benennungspraktiken für Aktionen, einschließlich Einschränkungen, finden Sie unter [Aktionsname](#).

Entsprechende Benutzeroberfläche: visueller Editor/ ***Aktionsname /Registerkarte Konfiguration/ Aktionsname*** oder Aktionsanzeigenname

action-group-name

(Actions/*action-group-name*)

(Optional)

Eine Aktionsgruppe enthält eine oder mehrere Aktionen. Das Gruppieren von Aktionen in Aktionsgruppen hilft Ihnen dabei, Ihren Arbeitsablauf zu organisieren, und ermöglicht es Ihnen auch, Abhängigkeiten zwischen verschiedenen Aktionsgruppen zu konfigurieren.

*action-group-name* Ersetzen Sie es durch einen Namen, den Sie der Aktionsgruppe geben möchten. Namen von Aktionsgruppen müssen innerhalb des Workflows eindeutig sein und dürfen nur alphanumerische Zeichen, Bindestriche und Unterstriche enthalten. Weitere Informationen zu Syntaxregeln finden Sie unter [Richtlinien für die YAML-Syntax](#)

Weitere Informationen zu Aktionsgruppen finden Sie unter [Gruppierung von Aktionen in Aktionsgruppen](#).

Entsprechende Benutzeroberfläche: keine

## Aktionsreferenz zum Erstellen und Testen

Im Folgenden finden Sie die YAML-Referenz zur Aktionsdefinition für Build- und Testaktionen. Es gibt eine Referenz für zwei Aktionen, da sich ihre YAML-Eigenschaften sehr ähnlich sind.

Wählen Sie im folgenden Code eine YAML-Eigenschaft aus, um eine Beschreibung zu erhalten.



**Note**

Die meisten der folgenden YAML-Eigenschaften haben entsprechende Benutzeroberflächenelemente im visuellen Editor. Verwenden Sie Strg+F, um nach einem UI-Element zu suchen. Das Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau 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"

```

## Aktionsname

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zuzulassen.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Aktionsname“

## Identifizier

*(Aktionsname/)* Identifizier

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nicht, es sei denn, Sie möchten die Version ändern. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Wird `aws/build@v1` für Build-Aktionen verwendet.

`aws/managed-test@v1` Für Testaktionen verwenden.

*Entsprechende Benutzeroberfläche: Workflow-Diagramm/Aktionsname/aws/build@v1 | aws/managed-test@v1 label*

## DependsOn

*(Aktionsname/)* **DependsOn**

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Hinweise zur Funktion „Hängt davon ab“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#)

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“/„Hängt davon ab“ — optional

## Compute

*(Aktionsname/)* Compute

(Optional)

Die Rechen-Engine, mit der Ihre Workflow-Aktionen ausgeführt wurden. Sie können die Berechnung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, jedoch nicht auf beiden Ebenen. Wenn auf Workflow-Ebene angegeben, gilt die Rechenkonfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instanz ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

(Typ des **Aktionsnamens**/Compute/)

(Erforderlich, wenn es enthalten [Compute](#) ist)

Der Typ der Compute Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsläufen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Berechnungstyp“

## Fleet

(/Compute/**FleetAktionsname**)

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden sollen. Bei bedarfsgesteuerten Flotten stellt der Workflow zu Beginn einer Aktion die benötigten Ressourcen bereit, und die Maschinen werden zerstört, wenn die Aktion abgeschlossen ist. Beispiele für Flotten auf Abruf: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Weitere Informationen zu Flotten auf Abruf finden Sie unter [Flotteneigenschaften auf Abruf](#)

Bei bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen, um Ihre Workflow-Aktionen auszuführen. Diese Maschinen bleiben im Leerlauf und sind bereit, Aktionen

sofort zu verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter.

[Bereitgestellte Flotteneigenschaften](#)

Wenn Fleet es weggelassen wird, ist die Standardeinstellung. `Linux.x86-64.Large`

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Flotte berechnen“

## Timeout

*(Aktionsname/)* Timeout

(Optional)

Geben Sie an, wie lange die Aktion in Minuten (YAML-Editor) oder Stunden und Minuten (Visual Editor) ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet wird. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Timeout“ — optional

## Environment

*(Aktionsname/)* Environment

(Optional)

Geben Sie die CodeCatalyst Umgebung an, die für die Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Umgebung“

## Name

*(/Environment/**Name**Aktionsname)*

(Optional)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/Name der Umgebung

## Connections

*(/Environment/ConnectionsAktionsname)*

(Optional)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können unter maximal eine Kontoverbindung angebenEnvironment.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen darüber, wie Sie eine Kontoverbindung mit Ihrer Umgebung verknüpfen, finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/

### Name

*(/Environment/Connections/NameAktionsname)*

(Optional)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/

### Role

*(/Environment/Connections/RoleAktionsname)*


(Optional)

Geben Sie den Namen der IAM-Rolle an, die diese Aktion verwendet, um auf AWS Dienste wie Amazon S3 und Amazon ECR zuzugreifen und diese zu nutzen. Stellen Sie sicher, dass diese Rolle zu Ihrer Kontoverbindung hinzugefügt wurde. Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

#### Note

Möglicherweise können Sie hier den Namen der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle angeben, sofern sie über ausreichende Berechtigungen verfügt. Weitere Informationen über diese Rolle

finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle über sehr umfangreiche Berechtigungen verfügt, was ein Sicherheitsrisiko darstellen kann. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger wichtig ist.

 Warning

Beschränken Sie die Berechtigungen auf diejenigen, die für die Build- und Testaktionen erforderlich sind. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/

## Caching

*(Aktionsname/)* Caching

(Optional)

Ein Abschnitt, in dem Sie einen Cache angeben können, um Dateien auf der Festplatte zu speichern und sie bei nachfolgenden Workflow-Ausführungen aus diesem Cache wiederherzustellen.

Weitere Informationen zum Zwischenspeichern von Dateien finden Sie unter [Arbeiten mit Datei-Caching](#)

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/Datei-Caching — optional

## FileCaching

*(Aktionsname) /Caching/ FileCaching*

(Optional)

Ein Abschnitt, der die Konfiguration für eine Sequenz von Caches spezifiziert.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Datei-Caching — optional/ Cache hinzufügen



## Schlüsselname-1

*(Aktionsname Schlüsselname-1) /Caching/FileCaching/*

(Optional)

Geben Sie den Namen Ihrer primären Cache-Eigenschaft an. Die Namen der Cache-Eigenschaften müssen innerhalb Ihres Workflows eindeutig sein. Jede Aktion kann bis zu fünf Einträge enthaltenFileCaching.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Datei-Caching — optional/Cache hinzufügen/Schlüssel

## Path

*(/Caching/FileCaching/PathAktionsname Schlüsselname-1/)*

(Optional)

Geben Sie den zugehörigen Pfad für Ihren Cache an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Datei-Caching — optional/Cache hinzufügen/Pfad

## RestoreKeys

*(/Caching/FileCaching/RestoreKeysAktionsname Schlüsselname-1/)*

(Optional)

Geben Sie den Wiederherstellungsschlüssel an, der als Fallback verwendet werden soll, wenn die primäre Cache-Eigenschaft nicht gefunden werden kann. Die Namen der Wiederherstellungsschlüssel müssen innerhalb Ihres Workflows eindeutig sein. Jeder Cache kann bis zu fünf Einträge enthaltenRestoreKeys.


Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Datei-Caching — optional/Cache hinzufügen/Schlüssel wiederherstellen — optional

## Inputs

*(Aktionsname/) **Inputs***

(Optional)

Der Inputs Abschnitt definiert die Daten, die eine Aktion während einer Workflow-Ausführung benötigt.

 Note

Pro Build-Aktion oder Testaktion sind maximal vier Eingaben (eine Quelle und drei Artefakte) zulässig. Variablen werden auf diese Summe nicht angerechnet.

Wenn Sie auf Dateien verweisen müssen, die sich in unterschiedlichen Eingaben befinden (z. B. eine Quelle und ein Artefakt), ist die Quelleingabe die primäre Eingabe und das Artefakt die sekundäre Eingabe. Verweise auf Dateien in sekundären Eingaben benötigen ein spezielles Präfix, um sie von der primären Eingabe zu unterscheiden. Details hierzu finden Sie unter [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#).

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“

## Sources

*(Aktionsname /Inputs/ Sources)*

(Optional)

Geben Sie die Labels an, die die Quell-Repositorys repräsentieren, die für die Aktion benötigt werden. Derzeit wird nur die Bezeichnung, die das Quell-Repository darstellt `WorkflowSource`, in dem Ihre Workflow-Definitionsdatei gespeichert ist, unterstützt.

Wenn Sie eine Quelle weglassen, müssen Sie mindestens ein Eingabeartefakt unter angeben.

*action-name/Inputs/Artifacts*

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: keine

## Artifacts - input

*(Aktionsname /Inputs/ Artifacts)*

(Optional)

Geben Sie Artefakte aus früheren Aktionen an, die Sie als Eingabe für diese Aktion bereitstellen möchten. Diese Artefakte müssen bereits in früheren Aktionen als Ausgabeartefakte definiert sein.

Wenn Sie keine Eingabeartefakte angeben, müssen Sie mindestens ein Quell-Repository unter angeben *action-name*/Inputs/Sources.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

#### Note

Wenn die Dropdownliste Artefakte — optional nicht verfügbar ist (visueller Editor) oder wenn Sie bei der Validierung Ihres YAML (YAML-Editors) Fehler erhalten, kann das daran liegen, dass die Aktion nur eine Eingabe unterstützt. Versuchen Sie in diesem Fall, die Quelleingabe zu entfernen.

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“/„Artefakte“ — optional

## Variables - input

*(/Inputs/VariablesAktionsname)*

(Optional)

Geben Sie eine Folge von Name/Wert-Paaren an, die die Eingabevariablen definieren, die Sie für die Aktion verfügbar machen möchten. Variablennamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Variablennamen zuzulassen.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“/„Variablen“ — optional

## Outputs

*(Aktionsname/)* Outputs

(Optional)

Definiert die Daten, die von der Aktion während einer Workflow-Ausführung ausgegeben werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“

## Artifacts - output

*(Aktionsname /Outputs/ **Artifacts**)*

(Optional)

Geben Sie den Namen eines Artefakts an, das durch die Aktion generiert wurde. Artefaktnamen müssen innerhalb eines Workflows eindeutig sein und sind auf alphanumerische Zeichen (a-z, A-Z, 0-9) und Unterstriche (\_) beschränkt. Leerzeichen, Bindestriche (-) und andere Sonderzeichen sind nicht zulässig. Sie können keine Anführungszeichen verwenden, um Leerzeichen, Bindestriche und andere Sonderzeichen in Namen von Ausgabeartefakten zuzulassen.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter. [Mit Artefakten arbeiten](#)

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Artefakte

## Name

*(/Outputs/Artifacts/**Name**Aktionsname)*

(Erforderlich, wenn es enthalten [Artifacts - output](#) ist)

Geben Sie den Namen eines Artefakts an, das durch die Aktion generiert wurde. Artefaktnamen müssen innerhalb eines Workflows eindeutig sein und sind auf alphanumerische Zeichen (a-z, A-Z, 0-9) und Unterstriche (\_) beschränkt. Leerzeichen, Bindestriche (-) und andere Sonderzeichen sind nicht zulässig. Sie können keine Anführungszeichen verwenden, um Leerzeichen, Bindestriche und andere Sonderzeichen in Namen von Ausgabeartefakten zuzulassen.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter. [Mit Artefakten arbeiten](#)

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben/Artefakte/Neue Ausgabe/Name des Artefakts erstellen“

## Files

*(Aktionsname) /Outputs/Artifacts/ **Files***

(Erforderlich, wenn es enthalten [Artifacts - output](#) ist)

Geben Sie die Dateien an, die in dem Artefakt CodeCatalyst enthalten sind, das von der Aktion ausgegeben wird. Diese Dateien werden durch die Workflow-Aktion generiert, wenn sie ausgeführt wird, und sind auch in Ihrem Quell-Repository verfügbar. Dateipfade können sich in einem Quell-Repository oder einem Artefakt aus einer früheren Aktion befinden und sind relativ zum Quell-Repository oder Artefakt-Stamm. Sie können Glob-Muster verwenden, um Pfade anzugeben.

Beispiele:

- Um eine einzelne Datei anzugeben, die sich im Stammverzeichnis Ihres Build-Speicherorts oder Quell-Repository-Speicherorts befindet, verwenden Sie `my-file.jar`.
- Um eine einzelne Datei in einem Unterverzeichnis anzugeben, verwenden Sie `directory/my-file.jar` oder `directory/subdirectory/my-file.jar`.
- Um alle Dateien anzugeben, verwenden Sie `**/*`. Das `**` Glob-Muster gibt an, dass es einer beliebigen Anzahl von Unterverzeichnissen entspricht.
- Um alle Dateien und Verzeichnisse in einem Verzeichnis mit dem Namen `directory`, verwenden Sie `directory/**/*`. Das `**` Glob-Muster gibt an, dass es einer beliebigen Anzahl von Unterverzeichnissen entspricht.
- Um alle Dateien in einem Verzeichnis mit dem Namen `directory`, aber nicht in einem seiner Unterverzeichnisse anzugeben, verwenden Sie `directory/*`.

#### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder ein anderes Sonderzeichen enthält, schließen Sie den Pfad in doppelte Anführungszeichen (") ein. "" Weitere Hinweise zu Sonderzeichen finden Sie unter [Richtlinien und Konventionen zur Syntax](#)

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

#### Note

Möglicherweise müssen Sie dem Dateipfad ein Präfix hinzufügen, um anzugeben, in welchem Artefakt oder in welcher Quelle das Objekt gefunden werden soll. Weitere Informationen finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Artefakte/Neue Ausgabe/ Von build erzeugte Dateien

## Variables - output

*(Aktionsname) /Outputs/ **Variables***

(Optional)

Geben Sie die Variablen an, die die Aktion exportieren soll, damit sie von nachfolgenden Aktionen verwendet werden können.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Variablen/Variable hinzufügen

## Variablenname-1

*(Aktionsname Variablenname-1) /Outputs/Variables/*

(Optional)

Geben Sie den Namen einer Variablen an, die von der Aktion exportiert werden soll. Diese Variable muss bereits im Steps Abschnitt Inputs oder derselben Aktion definiert sein.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Variablen/Variable hinzufügen/Name

## AutoDiscoverReports


*(Aktionsname) /Outputs/ **AutoDiscoverReports***

(Optional)

Definiert die Konfiguration für die Auto-Discovery-Funktion.

Wenn Sie die automatische Erkennung aktivieren, werden alle Inputs an die Aktion übergebenen sowie alle von der Aktion selbst generierten Dateien CodeCatalyst durchsucht, wobei nach Test-, Codeabdeckungs- und SCA-Berichten (Software Composition Analysis) gesucht wird. Wandelt jeden gefundenen Bericht in einen CodeCatalyst Bericht um. CodeCatalyst Ein CodeCatalyst Bericht ist ein

Bericht, der vollständig in den CodeCatalyst Service integriert ist und über die Konsole angezeigt und bearbeitet werden kann. CodeCatalyst

 Note

Standardmäßig überprüft die automatische Erkennungsfunktion alle Dateien. Mithilfe der Eigenschaften [IncludePaths](#) oder [ExcludePaths](#) können Sie einschränken, welche Dateien überprüft werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben/Berichte“/„Automatische Erkennung von Berichten“

Enabled

*(Aktionsname) /Outputs/AutoDiscoverReports/ **Enabled***

(Optional)

Aktivieren oder deaktivieren Sie die automatische Erkennungsfunktion.

Gültige Werte sind `true` oder `false`.

Wenn `Enabled` es weggelassen wird, ist `true` die Standardeinstellung.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben/Berichte“/„Automatische Erkennung von Berichten“

ReportNamePrefix

*(Aktionsname) /Outputs/AutoDiscoverReports/ **ReportNamePrefix***

(Erforderlich, wenn [AutoDiscoverReports](#) es enthalten und aktiviert ist)

Geben Sie ein Präfix an, das allen gefundenen Berichten CodeCatalyst vorangestellt wird, um die zugehörigen CodeCatalyst Berichte zu benennen. Wenn Sie beispielsweise das Präfix von `AutoDiscovered` angeben und zwei Testberichte CodeCatalyst automatisch erkennen, erhalten die zugehörigen CodeCatalyst Berichte den Namen `TestSuiteOne.xml` `AutoDiscoveredTestSuiteOne` und `TestSuiteTwo.xml`. `AutoDiscoveredTestSuiteTwo`

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Berichte/Präfixname

## IncludePaths

*(Aktionsname) /Outputs/AutoDiscoverReports/ **IncludePaths***

Oder

*(Aktionsname Berichtsname-1//Outputs/Reports/) **IncludePaths***

(Erforderlich, wenn [AutoDiscoverReports](#) es enthalten und aktiviert ist, oder wenn es enthalten ist)  
[Reports](#)

Geben Sie die Dateien und Dateipfade an, CodeCatalyst die bei der Suche nach Rohberichten berücksichtigt werden. Wenn Sie beispielsweise angeben `/test/report/*`, wird das gesamte [Build-Image](#), das von der Aktion verwendet wurde, nach dem `/test/report/*` Verzeichnis CodeCatalyst durchsucht. Wenn das Verzeichnis gefunden CodeCatalyst wird, wird in diesem Verzeichnis nach Berichten gesucht.

### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder andere Sonderzeichen enthält, schließen Sie den Pfad in doppelte Anführungszeichen (") ein. "" Weitere Hinweise zu Sonderzeichen finden Sie unter. [Richtlinien und Konventionen zur Syntax](#)

Wenn diese Eigenschaft weggelassen wird, ist die Standardeinstellung `**/*`, was bedeutet, dass die Suche alle Dateien in allen Pfaden umfasst.

### Note

Bei manuell konfigurierten Berichten `IncludePaths` muss es sich um ein Glob-Muster handeln, das einer einzelnen Datei entspricht.

Entsprechende Benutzeroberfläche:

- Registerkarte „Ausgaben/Berichte automatisch erkennen/Pfade einschließen/ausschließen/Pfade einschließen“



- Registerkarte „Ausgaben/Berichte/Berichte manuell konfigurieren/Berichtsname-1 /Pfade einschließen/ausschließen/ Pfade **einschließen**“

## ExcludePaths

*(/Outputs/AutoDiscoverReports/ExcludePathsAktionsname)*

Oder

*(Aktionsname Berichtsname-1//Outputs/Reports/) ExcludePaths*

(Optional)

Geben Sie die Dateien und Dateipfade an, die bei der Suche nach Rohberichten CodeCatalyst ausgeschlossen werden sollen. Wenn Sie beispielsweise angeben `/test/my-reports/**/*`, CodeCatalyst wird nicht nach Dateien im `/test/my-reports/` Verzeichnis gesucht. Um alle Dateien in einem Verzeichnis zu ignorieren, verwenden Sie das `**/*` Glob-Muster.

### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder andere Sonderzeichen enthält, schließen Sie den Pfad in doppelte Anführungszeichen (") ein. "" Weitere Hinweise zu Sonderzeichen finden Sie unter [Richtlinien und Konventionen zur Syntax](#)

Entsprechende Benutzeroberfläche:

- Registerkarte „Ausgaben/Berichte/Berichte automatisch erkennen/Pfade einschließen/ausschließen/Pfade ausschließen“
- Registerkarte „Ausgaben/Berichte/Berichte manuell konfigurieren/Berichtsname-1 /Pfade einschließen/ausschließen/ Pfade **ausschließen**“

## SuccessCriteria

*(/Outputs/AutoDiscoverReports/SuccessCriteriaAktionsname)*

Oder

*(Aktionsname Berichtsname-1//Outputs/Reports/) SuccessCriteria*

(Optional)

Geben Sie die Erfolgskriterien für die Berichte Test, Codeabdeckung, Software Composition Analysis (SCA) und Static Analysis (SA) an.

Weitere Informationen finden Sie unter [Erfolgskriterien für Berichte konfigurieren](#).

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien“

PassRate

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/ **PassRate***

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/ **PassRate***

(Optional)

Geben Sie den Prozentsatz der Tests in einem Testbericht an, die bestanden werden müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Zu den gültigen Werten gehören Dezimalzahlen. Zum Beispiel: 50, 60.5. Die Kriterien für die Erfolgsquote werden nur auf Testberichte angewendet. Weitere Informationen zu Testberichten finden Sie unter [Testberichte](#).

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien/Erfolgsquote“

LineCoverage

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/ **LineCoverage***

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
**LineCoverage***

(Optional)

Geben Sie den Prozentsatz der Zeilen in einem Bericht über die Codeabdeckung an, die abgedeckt werden müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Zu den gültigen Werten gehören Dezimalzahlen. Zum Beispiel: 50, 60.5. Die Kriterien für die

Leitungsabdeckung werden nur auf Berichte über die Codeabdeckung angewendet. Weitere Informationen zu Berichten über die Codeabdeckung finden Sie unter [Berichte zur Codeabdeckung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien“/  
„Linienabdeckung“

## BranchCoverage

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/ **BranchCoverage***

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
**BranchCoverage***

(Optional)

Geben Sie den Prozentsatz der Zweige in einem Bericht zur Codeabdeckung an, die abgedeckt werden müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Gültige Werte beinhalten Dezimalzahlen. Zum Beispiel: 50, 60.5. Die Kriterien für die Abdeckung von Filialen werden nur auf Berichte zur Codeabdeckung angewendet. Weitere Informationen zu Berichten über die Codeabdeckung finden Sie unter [Berichte zur Codeabdeckung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien“/  
„Branch Coverage“

## Vulnerabilities

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/ **Vulnerabilities***

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
**Vulnerabilities***

(Optional)

Geben Sie die maximale Anzahl und den Schweregrad der Sicherheitslücken an, die im SCA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert werden kann. Um Sicherheitslücken zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Sicherheitsanfälligkeiten, die Sie in die Anzahl einbeziehen möchten. Gültige Werte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählen HIGHHIGH, werden alle CRITICAL Sicherheitslücken gezählt.

- Die maximale Anzahl von Sicherheitslücken mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Kriterien für Sicherheitslücken werden nur auf SCA-Berichte angewendet. Weitere Informationen zu SCA-Berichten finden Sie unter [Berichte zur Analyse der Softwarezusammensetzung](#).

Verwenden Sie die `Severity` Eigenschaft, um den Mindestschweregrad anzugeben. Verwenden Sie die `Number` Eigenschaft, um die maximale Anzahl von Sicherheitslücken anzugeben.

Entsprechende Benutzeroberfläche: Registerkarte Ausgabe/Berichte/Erfolgskriterien/  
Sicherheitslücken

## StaticAnalysisBug

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/  
StaticAnalysisBug*

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
StaticAnalysisBug*

(Optional)

Geben Sie die maximale Anzahl und den Schweregrad der Fehler an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Um Fehler anzugeben, müssen Sie Folgendes angeben:

- Der minimale Schweregrad der Fehler, die Sie in die Zählung einbeziehen möchten. Gültige Werte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählen HIGH, dann werden HIGH alle CRITICAL Bugs gezählt.

- Die maximale Anzahl von Fehlern mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Bug-Kriterien werden nur auf PyLint ESLint SA-Berichte angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

Verwenden Sie die `Severity` Eigenschaft, um den Mindestschweregrad anzugeben. Verwenden Sie die `Number` Eigenschaft, um die maximale Anzahl von Sicherheitslücken anzugeben.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien“/„Bugs“

## StaticAnalysisSecurity

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/  
StaticAnalysisSecurity*

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
StaticAnalysisSecurity*

(Optional)

Geben Sie die maximale Anzahl und den Schweregrad der Sicherheitslücken an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert werden kann. Um Sicherheitslücken zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Sicherheitslücken, die Sie in die Zählung einbeziehen möchten. Gültige Werte (vom höchsten bis zum geringsten Schweregrad) sind: CRITICAL,HIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählen HIGH, dann HIGH werden CRITICAL Sicherheitslücken gezählt.

- Die maximale Anzahl von Sicherheitslücken mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Kriterien für Sicherheitslücken werden nur auf PyLint Berichte von ESLint SA angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

Verwenden Sie die `Severity` Eigenschaft, um den Mindestschweregrad anzugeben. Verwenden Sie die `Number` Eigenschaft, um die maximale Anzahl von Sicherheitslücken anzugeben.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien“/  
Sicherheitslücken

## StaticAnalysisQuality

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/  
**StaticAnalysisQuality***

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
**StaticAnalysisQuality***

(Optional)

Geben Sie die maximale Anzahl und den Schweregrad von Qualitätsproblemen an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert werden kann. Um Qualitätsprobleme zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Qualitätsprobleme, die Sie in die Zählung einbeziehen möchten. Gültige Werte (vom höchsten bis zum geringsten Schweregrad) sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie sich beispielsweise dafür entscheiden HIGHHIGH, werden CRITICAL Qualitätsprobleme gezählt.

- Die maximale Anzahl von Qualitätsproblemen mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Kriterien für Qualitätsprobleme werden nur auf PyLint Berichte von ESLint SA angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

Verwenden Sie die `Severity` Eigenschaft, um den Mindestschweregrad anzugeben. Verwenden Sie die `Number` Eigenschaft, um die maximale Anzahl von Sicherheitslücken anzugeben.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien“/  
„Qualitätsprobleme“

## StaticAnalysisFinding

*(Aktionsname) /Outputs/AutoDiscoverReports/SuccessCriteria/  
StaticAnalysisFinding*

Oder

*(Aktionsname Berichtsname-1/Outputs/Reports/) /SuccessCriteria/  
StaticAnalysisFinding*

(Optional)

Geben Sie die maximale Anzahl und den Schweregrad der Ergebnisse an, die im SA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert werden kann. Um Ergebnisse zu spezifizieren, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Ergebnisse, die Sie in die Zählung einbeziehen möchten. Gültige Werte, vom höchsten bis zum geringsten Schweregrad, sind: CRITICALHIGH,MEDIUM,,LOW,INFORMATIONAL.

Wenn Sie beispielsweise wählenHIGH, dann werden HIGH die CRITICAL Ergebnisse zusammengezählt.

- Die maximale Anzahl von Ergebnissen mit dem angegebenen Schweregrad, die Sie zulassen möchten. Wenn diese Zahl überschritten wird, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Die Ergebnisse werden nur auf Berichte der SARIF SA angewendet. Weitere Informationen zu SA-Berichten finden Sie unter [Statische Analyseberichte](#).

Verwenden Sie die `Severity` Eigenschaft, um den Mindestschweregrad anzugeben. Verwenden Sie die `Number` Eigenschaft, um die maximale Anzahl von Sicherheitslücken anzugeben.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgabe/Berichte/Erfolgskriterien/Ergebnisse“

## Reports

*/Outputs/Reports*(Name der Aktion)

(Optional)

Ein Abschnitt, der die Konfiguration für Testberichte spezifiziert.

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Berichte

Berichtsname-1

***(Aktionsname Berichtsname-1) /Outputs/Reports/***

(Erforderlich, wenn es enthalten ist) [Reports](#)

Der Name, den Sie dem CodeCatalyst Bericht geben möchten, der aus Ihren Rohberichten generiert wird.

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Berichte/Berichte manuell konfigurieren/  
Berichtsname

Format

***(/Outputs/Reports/Aktionsname **Format**Berichtsname-1/)***

(Erforderlich, falls enthalten) [Reports](#)

Geben Sie das Dateiformat an, das Sie für Ihre Berichte verwenden. Mögliche Werte können sein wie folgt.

- Für Testberichte:
  - Geben Sie für Cucumber JSON Cucumber (visueller Editor) oder CUCUMBERJSON (YAML-Editor) an.
  - Geben Sie für JUnit-XML JUnit (visueller Editor) oder JUNITXML (YAML-Editor) an.
  - Geben Sie für NUnit-XML NUnit (visueller Editor) oder NUNITXML (YAML-Editor) an.
  - Geben Sie für NUnit 3-XML NUnit3 (visueller Editor) oder (YAML-Editor) an. NUNIT3XML
  - Geben Sie für Visual Studio TRX Visual Studio TRX (visueller Editor) oder (YAML-Editor) an. VISUALSTUDIOTRX
  - Geben Sie für TestNG XML TestNG (visueller Editor) oder TESTNGXML (YAML-Editor) an.
- Für Berichte über die Codeabdeckung:
  - Geben Sie für Clover-XML Clover (visueller Editor) oder CLOVERXML (YAML-Editor) an.
  - Geben Sie für Cobertura XML Cobertura (visueller Editor) oder (YAML-Editor) an. COBERTURAXML
  - Geben Sie für JaCoCo XML JaCoCo(visueller Editor) oder JACOCOXML (YAML-Editor) an.



- Geben Sie für SimpleCov JSON, das von [simplecov und nicht von simplecov-json generiert wurde](#), [Simplecov](#) (visueller Editor) oder (YAML-Editor) an. SIMPLECOV
- Für SCA-Berichte (Software Composition Analysis):
  - Geben Sie für SARIF SARIF (Visual Editor) oder SARIFSCA (YAML-Editor) an.

**Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Berichte/Berichte manuell konfigurieren/Berichte hinzufügen/konfigurieren/Berichtsname-1 / Berichtstyp und Berichtsformat**

## Configuration

*(Aktionsname/) Configuration*

(Erforderlich) Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“

## Container

*(Aktionsname /Configuration/ Container)*

(Optional)

Geben Sie das Docker-Image oder den Container an, den die Aktion verwendet, um ihre Verarbeitung abzuschließen. Sie können eines der mitgelieferten [aktiven Images](#) angeben CodeCatalyst, oder Sie können Ihr eigenes Image verwenden. Wenn Sie Ihr eigenes Image verwenden möchten, kann es sich in Amazon ECR, Docker Hub oder einer anderen Registrierung befinden. Wenn Sie kein Docker-Image angeben, verwendet die Aktion eines der aktiven Images für die Verarbeitung. Informationen darüber, welches aktive Image standardmäßig verwendet wird, finden Sie unter [Aktive Bilder](#).

Weitere Informationen zur Angabe Ihres eigenen Docker-Images finden Sie unter [Zuweisen eines Docker-Images für eine benutzerdefinierte Laufzeitumgebung zu einer Aktion](#).

Entsprechende Benutzeroberfläche: Docker-Image für die Laufzeitumgebung — optional

## Registry

*(/Configuration/Container/RegistryAktionsname)*

(Erforderlich, wenn es enthalten Container ist)

Geben Sie die Registrierung an, in der Ihr Bild gespeichert ist. Gültige Werte sind:

- CODECATALYST(YAML-Editor)

Das Bild wird in der CodeCatalyst Registrierung gespeichert.

- Docker Hub (visueller Editor) oder DockerHub (YAML-Editor)

Das Bild wird in der Docker Hub-Image-Registry gespeichert.

- Andere Registrierung (visueller Editor) oder Other (YAML-Editor)

Das Bild wird in einer benutzerdefinierten Bildregistrierung gespeichert. Jede öffentlich verfügbare Registrierung kann verwendet werden.

- Amazon Elastic Container Registry (visueller Editor) oder ECR (YAML-Editor)

Das Bild wird in einem Image-Repository der Amazon Elastic Container Registry gespeichert.

Um ein Bild in einem Amazon ECR-Repository zu verwenden, benötigt diese Aktion Zugriff auf Amazon ECR. Um diesen Zugriff zu aktivieren, müssen Sie eine [IAM-Rolle](#) erstellen, die die folgenden Berechtigungen und eine benutzerdefinierte Vertrauensrichtlinie umfasst. (Sie können eine bestehende Rolle so ändern, dass sie die Berechtigungen und die Richtlinie einbezieht, wenn Sie möchten.)

Die IAM-Rolle muss die folgenden Berechtigungen in ihrer Rollenrichtlinie enthalten:

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

Die IAM-Rolle muss die folgende benutzerdefinierte Vertrauensrichtlinie enthalten:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```
        "codecatalyst-runner.amazonaws.com",
        "codecatalyst.amazonaws.com"
    ],
    },
    "Action": "sts:AssumeRole"
}
]
```

Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter [Erstellen einer Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien \(Konsole\)](#) im IAM-Benutzerhandbuch.

Nachdem Sie die Rolle erstellt haben, müssen Sie sie der Aktion über eine Umgebung zuweisen. Weitere Informationen finden Sie unter [Zuordnen einer Umgebung, einer Kontoverbindung und einer IAM-Rolle zu einer Workflow-Aktion](#).

Entsprechende Benutzeroberfläche: Amazon Elastic Container Registry, Docker Hub und andere Registrierungsoptionen

## Image

(*/Configuration/Container/ImageAktionsname*)

(Erforderlich, wenn es enthalten Container ist)

Geben Sie eines der folgenden Elemente an:

- Wenn Sie eine CODECATALYST Registrierung verwenden, legen Sie für das Image eines der folgenden [aktiven Images](#) fest:
  - 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

- Wenn Sie eine Docker Hub-Registry verwenden, legen Sie für das Image den Namen des Docker Hub-Images und das optionale Tag fest.

Beispiel: `postgres:latest`

- Wenn Sie eine Amazon ECR-Registrierung verwenden, setzen Sie das Bild auf die Amazon ECR-Registrierungs-URI.

Beispiel: `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

- Wenn Sie eine benutzerdefinierte Registrierung verwenden, setzen Sie das Bild auf den Wert, der von der benutzerdefinierten Registrierung erwartet wird.

Entsprechende Benutzeroberfläche: Docker-Image der Laufzeitumgebung (falls die Registrierung vorhanden ist **CODECATALYST**), Docker Hub-Image (wenn die Registrierung Docker Hub ist), ECR-Image-URL (wenn die Registrierung Amazon Elastic Container Registry ist) und Image-URL (wenn die Registrierung Andere Registrierung ist).

## Steps

*(/Configuration/StepsAktionsname)*

(Erforderlich)

Geben Sie die Shell-Befehle an, die Sie während der Aktion zur Installation, Konfiguration und Ausführung Ihrer Build-Tools ausführen möchten.

Hier ist ein Beispiel für die Erstellung eines NPM-Projekts:

Steps:

- Run: `npm install`
- Run: `npm run build`

Hier ist ein Beispiel für die Angabe von Dateipfaden:

Steps:

- Run: `cd $ACTION_BUILD_SOURCE_PATH_WorkflowSource/app && cat file2.txt`
- Run: `cd $ACTION_BUILD_SOURCE_PATH_MyBuildArtifact/build-output/ && cat file.txt`

Weitere Hinweise zur Angabe von Dateipfaden finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/Shell-Befehle

## Packages

*(/Configuration/PackagesAktionsname)*

(Optional)

Ein Abschnitt, in dem Sie ein Paket-Repository angeben können, das die Aktion zum Auflösen von Abhängigkeiten verwendet. Mit Paketen können Sie Softwarepakete, die für die Anwendungsentwicklung verwendet werden, sicher speichern und gemeinsam nutzen.

Weitere Informationen zu Paketen finden Sie unter [Pakete in CodeCatalyst](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Pakete“

## NpmConfiguration

*(/Configuration/Packages/NpmConfigurationAktionsname)*

(Erforderlich, wenn es enthalten [Packages](#) ist)

Ein Abschnitt, der die Konfiguration für das npm-Paketformat definiert. Diese Konfiguration wird von einer Aktion während einer Workflow-Ausführung verwendet.

Weitere Informationen zur Konfiguration des npm-Pakets finden Sie unter [Verwenden von npm](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Pakete/Konfiguration hinzufügen/  
npm

## PackageRegistries

*(Aktionsname) /Configuration/Packages/NpmConfiguration/ PackageRegistries*

(Erforderlich, wenn es enthalten [Packages](#) ist)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften einer Folge von Paket-Repositorys definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Pakete/Konfiguration hinzufügen/  
npm/Paket-Repository hinzufügen

## PackagesRepository

*(Aktionsname) /Configuration/Packages/NpmConfiguration/PackageRegistries/  
**PackagesRepository***

(Erforderlich, wenn es enthalten [Packages](#) ist)

Geben Sie den Namen Ihres CodeCatalyst Paket-Repositorys an, das die Aktion verwenden soll.

Wenn Sie mehrere Standard-Repositorys angeben, hat das letzte Repository Priorität.

Weitere Hinweise zu Paket-Repositorys finden Sie unter [Paket-Repositorys](#)

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Pakete/Konfiguration hinzufügen/  
npm/Paket-Repository hinzufügen/Paket-Repository

## Scopes

*(Aktionsname) /Configuration/Packages/NpmConfiguration/PackageRegistries/  
**Scopes***

(Optional)

Geben Sie eine Abfolge von Bereichen an, die Sie in Ihrer Paketregistrierung definieren möchten. Bei der Definition von Bereichen wird das angegebene Paket-Repository als Registrierung für alle aufgelisteten Bereiche konfiguriert. Wenn ein Paket mit dem Bereich über den npm-Client angefordert wird, verwendet es dieses Repository anstelle des Standard-Repositorys. Jedem Bereichsnamen muss ein „@“ vorangestellt werden.

Wenn Sie übergeordnete Bereiche einbeziehen, hat das letzte Repository Vorrang.

Wenn Scopes es weggelassen wird, wird das angegebene Paket-Repository als Standardregistrierung für alle von der Aktion verwendeten Pakete konfiguriert.

Weitere Informationen zu Bereichen finden Sie unter [Paket-Namespaces](#) und Pakete mit [Geltungsbereich](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Pakete/Konfiguration hinzufügen/  
npm/Paket-Repository hinzufügen/Bereiche — optional

## Referenz zur Aktion „Amazon S3 veröffentlichen“

Im Folgenden finden Sie die YAML-Referenz für die Aktionsdefinition für die Amazon S3-Veröffentlichungsaktion. Informationen zur Verwendung dieser Aktion finden Sie unter [Die Aktion „Amazon S3 S3-Veröffentlichung“ hinzufügen](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Um ein UI-Element nachzuschlagen, verwenden Sie Strg+F . Das -Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau 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: s3-bucket-name  
  TargetPath: my/target
```

## S3Publish

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: S3Publish\_nn.

Entsprechende Benutzeroberfläche: Registerkarte/Aktionsname der Konfiguration

### Identifizier

(*S3Publish*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: aws/s3-publish@v1.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/S3Publish\_nn/aws/s3-publish@v1-Bezeichnung

### DependsOn

(*S3Publish*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.



Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

(*S3Publish*/Compute)

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

(*S3Publish*/Compute/Type)

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsausführungen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Datenverarbeitungstyp „Konfiguration“

## Fleet

(*S3Publish*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, die Ihren Workflow oder Ihre Workflow-Aktionen ausführen wird. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen `Fleet` wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Datenverarbeitungsflotte

## Timeout

(*S3Publish*/Timeout)

(Erforderlich)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Inputs

(*S3Publish*/Inputs)

(Optional)

Der `Inputs` Abschnitt definiert die Daten, die der während einer Workflow-Ausführung `S3Publish` benötigt.

**Note**

Für jede AWS CDK Bereitstellungsaktion sind maximal vier Eingaben (eine Quelle und drei Artefakte) zulässig. Variablen werden nicht auf diese Summe angerechnet.

Wenn Sie auf Dateien verweisen müssen, die sich in verschiedenen Eingaben befinden (z. B. eine Quelle und ein Artefakt), ist die Quelleingabe die primäre Eingabe und das Artefakt ist die sekundäre Eingabe. Verweise auf Dateien in sekundären Eingaben erfordern ein spezielles Präfix, um sie vom primären zu trennen. Details hierzu finden Sie unter [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

## Sources

(*S3Publish*/Inputs/Sources)

(Erforderlich, wenn die Dateien, die Sie in Amazon S3 veröffentlichen möchten, in einem Quell-Repository gespeichert sind)

Wenn die Dateien, die Sie in Amazon S3 veröffentlichen möchten, in einem Quell-Repository gespeichert sind, geben Sie die Bezeichnung dieses Quell-Repositorys an. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`.

Wenn die Dateien, die Sie in Amazon S3 veröffentlichen möchten, nicht in einem Quell-Repository enthalten sind, müssen sie sich in einem Artefakt befinden, das von einer anderen Aktion generiert wird.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

(*S3Publish*/Inputs/Artifacts)

(Erforderlich, wenn die Dateien, die Sie in Amazon S3 veröffentlichen möchten, in einem [Ausgabeartefakt](#) aus einer vorherigen Aktion gespeichert sind)

Wenn die Dateien, die Sie in Amazon S3 veröffentlichen möchten, in einem Artefakt enthalten sind, das von einer vorherigen Aktion generiert wurde, geben Sie dieses Artefakt hier an. Wenn Ihre Dateien nicht in einem Artefakt enthalten sind, müssen sie sich in Ihrem Quell-Repository befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte der Konfiguration – optional

## Variables - input

(*S3Publish*/Inputs/Variables)

(Optional)

Geben Sie eine Sequenz von Name/Wert-Paaren an, die die Eingabevariablen definieren, die Sie der Aktion zur Verfügung stellen möchten. Variablennamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Variablennamen zu aktivieren.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Variablen Eingaben – optional

## Environment

(*S3Publish*/Environment)

(Erforderlich)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/„Umgebung/Verbindung/Rolle“/Umgebung

## Name

(*S3Publish*/Environment/Name)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/„Umgebung/Verbindung/  
Rolle“/Umgebung

## Connections

(*S3Publish*/Environment/Connections)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können maximal eine Kontoverbindung unter angebenEnvironment.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/ „Umgebung/Verbindung/  
Rolle“/Verbindung

## Name

(*S3Publish*/Environment/Connections/Name)

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/ „Umgebung/Verbindung/  
Rolle“/Verbindung


## Role

(*S3Publish*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die Amazon S3-Veröffentlichungsaktion verwendet, um auf Dateien zuzugreifen AWS und Dateien nach Amazon S3 zu kopieren. Stellen Sie sicher, dass diese Rolle Folgendes beinhaltet:

- Die folgende Berechtigungsrichtlinie:

 Warning

Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie aufgeführten. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
  "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/*"
      ]
    }
  ]
}
```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
    },
  ],
}
```

```
        "Action": "sts:AssumeRole"
    }
  ]
}
```

Stellen Sie sicher, dass diese Rolle mit Ihrer Kontoverbindung verknüpft ist. Weitere Informationen zum Zuordnen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

### Note

Sie können den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/ „Umgebung/Verbindung/Rolle“/Rolle

## Configuration

(*S3Publish*/Configuration)

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## SourcePath

(*S3Publish*/Configuration/SourcePath)

(Erforderlich)

Geben Sie den Namen und den Pfad eines Verzeichnisses oder einer Datei an, die Sie in Amazon S3 veröffentlichen möchten. Das Verzeichnis oder die Datei kann sich in einem Quell-Repository oder einem Artefakt aus einer vorherigen Aktion befinden und bezieht sich auf das Quell-Repository oder den Artefaktstamm.

Beispiele:

Durch die Angabe von `./myFolder` wird der Inhalt von `./myFolder` nach Amazon S3 `./myFolder/` kopiert und die zugrunde liegende Verzeichnisstruktur beibehalten.

Angabe von `./myFolder/myfile.txt` kopiert nur `myfile.txt` in Amazon S3. (Die Verzeichnisstruktur wird entfernt.)

Sie können keine Platzhalter verwenden.

#### Note

Möglicherweise müssen Sie dem Verzeichnis oder Dateipfad ein Präfix hinzufügen, um anzugeben, in welchem Artefakt oder welcher Quelle es gefunden werden soll. Weitere Informationen finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellpfad der Konfiguration

**DestinationBucketName**

*(S3Publish/Configuration/DestinationBucketName)*

(Erforderlich)

Geben Sie den Namen des Amazon S3-Buckets an, in dem Sie Dateien veröffentlichen möchten.

Entsprechende Benutzeroberfläche: Konfigurations-Registerkarte/Ziel-Bucket – optional

**TargetPath**

*(S3Publish/Configuration/TargetPath)*

(Optional)



Geben Sie den Namen und den Pfad des Verzeichnisses in Amazon S3 an, in dem Sie Ihre Dateien veröffentlichen möchten. Wenn das Verzeichnis nicht vorhanden ist, wird es erstellt. Der Verzeichnispfad darf den Bucket-Namen nicht enthalten.

Beispiele:

myS3Folder

./myS3Folder/myS3Subfolder

Entsprechende Benutzeroberfläche: Registerkarte/Zielverzeichnis Konfiguration – optional

## Referenz zur Aktion „AWS CDK bootstrap“

Im Folgenden finden Sie die YAML-Referenz der Aktionsdefinition für die AWS CDK Bootstrap-Aktion. Informationen zur Verwendung dieser Aktion finden Sie unter [Aktion „AWS CDK Bootstrap“ hinzufügen](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKBootstrapAction\_nn:
  Identifier: aws/cdk-bootstrap@v1
  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

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: `CDKBootstrapAction_nn`.

Entsprechende Benutzeroberfläche: Konfigurationsregisterkarte/Anzeigename der Aktion

### Identifizier

(*CDKBootstrapAction*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: `aws/cdk-bootstrap@v1`.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/CDKBootstrapAction\_nn/aws/cdk-bootstrap@v1-Bezeichnung

## DependsOn

(*CDKBootstrapAction*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

(*CDKBootstrapAction*/Compute)

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

(*CDKBootstrapAction*/Compute/Type)

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)

Optimiert für Flexibilität bei Aktionsausführungen.

- Lambda (visueller Editor) oder Lambda (YAML-Editor)

Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungstyp

## Fleet

(*CDKBootstrapAction*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, die Ihren Workflow oder Ihre Workflow-Aktionen ausführen wird. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen Fleet wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungsflotte

## Timeout

(*CDKBootstrapAction*/Timeout)

(Erforderlich)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Inputs

(*CDKBootstrapAction*/Inputs)

(Optional)

Der `Inputs` Abschnitt definiert die Daten, die die AWS CDK Bootstrap-Aktion während einer Workflow-Ausführung benötigt.

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

### Note

Für jede AWS CDK Bootstrap-Aktion ist nur eine Eingabe (entweder eine Quelle oder ein Artefakt) zulässig.

## Sources

(*CDKBootstrapAction*/Inputs/Sources)

(Erforderlich, wenn Ihre AWS CDK App in einem Quell-Repository gespeichert ist)

Wenn Ihre AWS CDK App in einem Quell-Repository gespeichert ist, geben Sie die Bezeichnung dieses Quell-Repositorys an. Die AWS CDK Bootstrap-Aktion synthetisiert die App in diesem Repository, bevor der Bootstrapping-Prozess gestartet wird. Derzeit ist die einzige unterstützte Repository-Bezeichnung `WorkflowSource`.

Wenn Ihre AWS CDK App nicht in einem Quell-Repository enthalten ist, muss sie sich in einem Artefakt befinden, das von einer anderen Aktion generiert wird.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

(*CDKBootstrapAction*/Inputs/Artifacts)

(Erforderlich, wenn Ihre AWS CDK App in einem [Ausgabeartefakt](#) einer vorherigen Aktion gespeichert ist)

Wenn Ihre AWS CDK App in einem Artefakt enthalten ist, das von einer vorherigen Aktion generiert wurde, geben Sie dieses Artefakt hier an. Die AWS CDK Bootstrap-Aktion synthetisiert die App im angegebenen Artefakt in eine CloudFormation Vorlage, bevor der Bootstrapping-Prozess gestartet wird. Wenn Ihre AWS CDK App nicht in einem Artefakt enthalten ist, muss sie sich in Ihrem Quell-Repository befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte – optional

## Outputs

(*CDKBootstrapAction*/Outputs)

(Optional)

Definiert die Daten, die von der Aktion während einer Workflow-Ausführung ausgegeben werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“

## Artifacts - output

(*CDKBootstrapAction*/Outputs/Artifacts)

(Optional)

Geben Sie die Artefakte an, die von der Aktion generiert wurden. Sie können diese Artefakte als Eingabe in anderen Aktionen referenzieren.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte „Outputs“/„Artifacts“

## Name

(*CDKBootstrapAction*/Outputs/Artifacts/Name)

(Erforderlich, wenn enthalten [Artifacts - output](#) ist)

Geben Sie den Namen des Artefakts an, das die AWS CloudFormation Vorlage enthält, die von der AWS CDK Bootstrap-Aktion zur Laufzeit synthetisiert wird. Der Standardwert ist `cdk_bootstrap_artifacts`. Wenn Sie kein Artefakt angeben, synthetisiert die Aktion die Vorlage,

speichert sie aber nicht in einem Artefakt. Erwägen Sie, die synthetisierte Vorlage in einem Artefakt zu speichern, um eine Aufzeichnung davon zu Test- oder Fehlerbehebungszwecken beizubehalten.

Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Artifacts/Artefakt hinzufügen/Artefakname erstellen

## Files

(*CDKBootstrapAction*/Outputs/Artifacts/Files)

(Erforderlich, wenn enthalten [Artifacts - output](#) ist)

Geben Sie die Dateien an, die in das Artefakt aufgenommen werden sollen. Sie müssen angeben "cdk.out/\*\*/\*", um die synthetisierte AWS CloudFormation Vorlage Ihrer AWS CDK App einzuschließen.

### Note

cdk.out ist das Standardverzeichnis, in dem synthetisierte Dateien gespeichert werden. Wenn Sie ein anderes Ausgabeverzeichnis als cdk.out in Ihrer -cdk.jsonDatei angegeben haben, geben Sie dieses Verzeichnis hier anstelle von ancdk.out.

Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Artifacts/Hinzufügen von Artefakten/Dateien, die vom Build erstellt wurden

## Environment

(*CDKBootstrapAction*/Environment)

(Erforderlich)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

## Name

(*CDKBootstrapAction*/Environment/Name)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

## Connections

(*CDKBootstrapAction*/Environment/Connections)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können maximal eine Kontoverbindung unter angebenEnvironment.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Name

(*CDKBootstrapAction*/Environment/Connections/Name)

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Role

(*CDKBootstrapAction*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die AWS CDK Bootstrap-Aktion für den Zugriff verwendet, AWS und fügen Sie den Bootstrap-Stack hinzu. Stellen Sie sicher, dass diese Rolle die folgenden Richtlinien enthält:



**Note**

Die in der folgenden Berechtigungsrichtlinie aufgeführten Berechtigungen sind diejenigen, die der `cdk bootstrap` Befehl benötigt, um sein Bootstrapping durchzuführen. Diese Berechtigungen können sich ändern, wenn der seinen Bootstrap-Befehl AWS CDK ändert.

**Warning**

Verwenden Sie diese Rolle nur mit der AWS CDK Bootstrap-Aktion . Es ist sehr zulässig und die Verwendung mit anderen Aktionen kann ein Sicherheitsrisiko darstellen.

- Die folgende Berechtigungsrichtlinie:

**Warning**

Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie aufgeführten. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "ssm:GetParameterHistory",
        "ecr:PutImageScanningConfiguration",
        "cloudformation:*",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "ssm:GetParameters",
        "iam:PutRolePolicy",
        "ssm:GetParameter",
        "ssm>DeleteParameters",
        "ecr>DeleteRepository",
```

```

        "ssm:PutParameter",
        "ssm>DeleteParameter",
        "iam:PassRole",
        "ecr:SetRepositoryPolicy",
        "ssm:GetParametersByPath",
        "ecr:DescribeRepositories",
        "ecr:GetLifecyclePolicy"
    ],
    "Resource": [
        "arn:aws:ssm:aws-region:aws-account:parameter/cdk-bootstrap/*",
        "arn:aws:cloudformation:aws-region:aws-account:stack/CDKToolkit/*",
        "arn:aws:ecr:aws-region:aws-account:repository/cdk-*",
        "arn:aws:iam::aws-account:role/cdk-*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:RegisterType",
        "cloudformation:CreateUploadBucket",
        "cloudformation:ListExports",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:SetTypeDefaultVersion",
        "cloudformation:RegisterPublisher",
        "cloudformation:ActivateType",
        "cloudformation:ListTypes",
        "cloudformation:DeactivateType",
        "cloudformation:SetTypeConfiguration",
        "cloudformation:DeregisterType",
        "cloudformation:ListTypeRegistrations",
        "cloudformation:EstimateTemplateCost",
        "cloudformation:DescribeAccountLimits",
        "cloudformation:BatchDescribeTypeConfigurations",
        "cloudformation:CreateStackSet",
        "cloudformation:ListStacks",
        "cloudformation:DescribeType",
        "cloudformation:ListImports",
        "s3:*",
        "cloudformation:PublishType",
        "ecr:CreateRepository",
        "cloudformation:DescribePublisher",
        "cloudformation:DescribeTypeRegistration",
        "cloudformation:TestType",
    ]
}

```

```

        "cloudformation:ValidateTemplate",
        "cloudformation:ListTypeVersions"
    ],
    "Resource": "*"
}
]
}

```

### Note

Wenn die Rolle zum ersten Mal verwendet wird, verwenden Sie den folgenden Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach ihrer Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Stellen Sie sicher, dass diese Rolle zu Ihrer Kontoverbindung hinzugefügt wird. Weitere Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

**Note**

Sie können den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Rolle

## Configuration

(*CDKBootstrapAction*/Configuration)

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## Region

(*CDKBootstrapAction*/Configuration/Region)

(Erforderlich)

Geben Sie die an, AWS-Region in der der Bootstrap-Stack bereitgestellt wird. Diese Region sollte mit der Region übereinstimmen, in der Ihre AWS CDK App bereitgestellt wird. Eine Liste der Regioncodes finden Sie unter [Regionale Endpunkte](#).

Entsprechende Benutzeroberfläche: Registerkarte/Region Konfiguration

## CdkCliVersion

(*CDKBootstrapAction*/Configuration/CdkCliVersion)

(Optional)

Diese Eigenschaft ist mit Version 1.0.13 oder höher der AWS CDK Bereitstellungsaktion und Version 1.0.8 oder höher der AWS CDK Bootstrap-Aktion verfügbar.

Geben Sie eines der folgenden Elemente an:

- Die vollständige Version der -AWS Cloud Development Kit (AWS CDK) Befehlszeilenschnittstelle (CLI) (auch AWS CDK Toolkit genannt), die diese Aktion verwenden soll. Beispiel: 2.102.1. Erwägen Sie, eine vollständige Version anzugeben, um Konsistenz und Stabilität beim Erstellen und Bereitstellen Ihrer Anwendung zu gewährleisten.

Oder

- latest. Erwägen Sie, anzugeben latest, um die neuesten Funktionen und Fehlerbehebungen der CDK-CLI zu nutzen.

Die Aktion lädt die angegebene Version (oder die neueste Version) der AWS CDK CLI in das CodeCatalyst [Build-Image](#) herunter und verwendet diese Version dann, um die Befehle auszuführen, die für die Bereitstellung Ihrer CDK-Anwendung oder das Bootstrapping Ihrer AWS Umgebung erforderlich sind.

Eine Liste der unterstützten CDK-CLI-Versionen, die Sie verwenden können, finden Sie unter [AWS CDK Versionen](#).

Wenn Sie diese Eigenschaft weglassen, verwendet die Aktion eine Standard-AWS CDK CLI-Version, die in einem der folgenden Themen beschrieben ist:

- [CDK-CLI-Versionen, die von der Aktion „AWS CDK Deploy“ verwendet werden](#)
- [CDK-CLI-Versionen, die von der Aktion „AWS CDK Bootstrap“ verwendet werden](#)

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/CLI AWS CDK-Version

## Referenz zur Aktion „AWS CDK Bereitstellen“

Im Folgenden finden Sie die YAML-Referenz zur Aktionsdefinition für die AWS CDK Bereitstellungsaktion. Informationen zur Verwendung dieser Aktion finden Sie unter [Aktion „AWS CDK Deploy“ hinzufügen](#).

**Note**

Die meisten der folgenden YAML-Eigenschaften haben entsprechende Benutzeroberflächenelemente im visuellen Editor. Verwenden Sie Strg+F, um nach einem UI-Element zu suchen. Das Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKDeploy\_nn:
  Identifier: aws/cdk-deploy@v1
  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/cdk.json  
CfnOutputVariables: '["CfnOutputKey1", "CfnOutputKey2", "CfnOutputKey3"]'  
CloudAssemblyRootPath: path-to-cdk.out
```

## CDKDeploy

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zuzulassen.

Standard: CDKDeploy\_nn.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Aktionsname“

### Identifizier

(*CDKDeploy*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: aws/cdk-deploy@v1.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/ CDKDeploy\_nn/ aws/cdk-deploy @v1 label

### DependsOn

(*CDKDeploy*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit die Bereitstellungsaktion ausgeführt werden kann. AWS CDK Wir empfehlen, die AWS CDK Bootstrap-Aktion in der DependsOn Eigenschaft wie folgt anzugeben:

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v1
  DependsOn:
    - CDKBootstrap
```

### Note

[Bootstrapping](#) ist eine zwingende Voraussetzung für die Bereitstellung einer App. AWS CDK Wenn Sie die AWS CDK Bootstrap-Aktion nicht in Ihren Workflow aufnehmen, müssen Sie eine andere Möglichkeit finden, den AWS CDK Bootstrap-Stack bereitzustellen, bevor Sie Ihre Bereitstellungsaktion ausführen. AWS CDK Weitere Informationen finden Sie unter [Voraussetzungen](#) in [Aktion „AWS CDK Deploy“ hinzufügen](#).

Weitere Informationen zur Funktion „Hängt davon ab“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#)

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“/„Hängt davon ab“ — optional

## Compute

(*CDKDeploy*/Compute)

(Optional)

Die Rechen-Engine, mit der Ihre Workflow-Aktionen ausgeführt wurden. Sie können die Berechnung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht beide. Wenn auf Workflow-Ebene angegeben, gilt die Rechenkonfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instanz ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

(*CDKDeploy*/Compute/Type)

([Compute](#)Erforderlich, falls enthalten)

Der Typ der Compute Engine. Sie können einen der folgenden Werte verwenden:



- EC2 (visueller Editor) oder EC2 (YAML-Editor)

Optimiert für Flexibilität bei Aktionsläufen.

- Lambda (visueller Editor) oder Lambda (YAML-Editor)

Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Erweitert — optional/  
Berechnungstyp

## Fleet

(*CDKDeploy*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden sollen. Bei bedarfsgesteuerten Flotten stellt der Workflow zu Beginn einer Aktion die benötigten Ressourcen bereit, und die Maschinen werden zerstört, wenn die Aktion abgeschlossen ist. Beispiele für Flotten auf Abruf: `Linux.x86-64.Large`, `Linux.x86-64.XLarge` Weitere Informationen zu Flotten auf Abruf finden Sie unter [Flotteneigenschaften auf Abruf](#)

Bei bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen, um Ihre Workflow-Aktionen auszuführen. Diese Maschinen bleiben im Leerlauf und sind bereit, Aktionen sofort zu verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#)

Wenn Fleet es weggelassen wird, ist die Standardeinstellung `Linux.x86-64.Large`

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Erweitert — optional/  
Compute Fleet

## Timeout

(*CDKDeploy*/Timeout)

(Erforderlich)

Geben Sie an, wie lange die Aktion in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) ausgeführt werden kann, bevor die Aktion beendet wird. CodeCatalyst Das Minimum beträgt

5 Minuten und das Maximum ist unter [beschrieben](#) [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Timeout“ — optional

## Inputs

(*CDKDeploy*/Inputs)

(Optional)

Der Inputs Abschnitt definiert die Daten, die während einer Workflow-Ausführung `CDKDeploy` benötigt werden.

### Note

Für jede AWS CDK Bereitstellungsaktion ist nur eine Eingabe (entweder eine Quelle oder ein Artefakt) zulässig.

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“

## Sources

(*CDKDeploy*/Inputs/Sources)

(Erforderlich, wenn die AWS CDK App, die Sie bereitstellen möchten, in einem Quell-Repository gespeichert ist)

Wenn Ihre AWS CDK App in einem Quell-Repository gespeichert ist, geben Sie die Bezeichnung dieses Quell-Repositorys an. Die AWS CDK Bereitstellungsaktion synthetisiert die App in diesem Repository, bevor der Bereitstellungsprozess gestartet wird. Derzeit ist `WorkflowSource` das einzige unterstützte Label.

Wenn Ihre AWS CDK App nicht in einem Quell-Repository enthalten ist, muss sie sich in einem Artefakt befinden, das durch eine andere Aktion generiert wurde.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“/„Quellen“ — optional

## Artifacts - input

(*CDKDeploy*/Inputs/Artifacts)

(Erforderlich, wenn die AWS CDK App, die Sie bereitstellen möchten, in einem [Ausgabeartefakt](#) einer vorherigen Aktion gespeichert ist)

Wenn Ihre AWS CDK App in einem Artefakt enthalten ist, das durch eine vorherige Aktion generiert wurde, geben Sie dieses Artefakt hier an. Die AWS CDK Bereitstellungsaktion synthetisiert die App im angegebenen Artefakt zu einer CloudFormation Vorlage, bevor der Bereitstellungsprozess gestartet wird. Wenn Ihre AWS CDK App nicht in einem Artefakt enthalten ist, muss sie sich in Ihrem Quell-Repository befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#)

Entsprechende Benutzeroberfläche: Registerkarte „Eingaben“/„Artefakte“ — optional

## Outputs

(*CDKDeploy*/Outputs)

(Optional)

Definiert die Daten, die von der Aktion während einer Workflow-Ausführung ausgegeben werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“

## Artifacts - output

(*CDKDeploy*/Outputs/Artifacts)

(Optional)

Geben Sie die durch die Aktion generierten Artefakte an. Sie können diese Artefakte als Eingabe in anderen Aktionen referenzieren.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Artefakte

## Name

(*CDKDeploy*/Outputs/Artifacts/Name)

(Erforderlich, wenn [Artifacts - output](#) es enthalten ist)

Geben Sie den Namen des Artefakts an, das die AWS CloudFormation Vorlage enthalten soll, die durch die AWS CDK Bereitstellungsaktion zur Laufzeit synthetisiert wird. Der Standardwert ist `cdk_artifact`. Wenn Sie kein Artefakt angeben, synthetisiert die Aktion die Vorlage, speichert sie jedoch nicht in einem Artefakt. Erwägen Sie, die synthetisierte Vorlage in einem Artefakt zu speichern, um sie zu Test- oder Fehlerbehebungszwecken aufzuzeichnen.

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Artefakte/Artefakt hinzufügen/Namen des Artefakts erstellen

## Files

(*CDKDeploy*/Outputs/Artifacts/Files)

([Artifacts - output](#) Erforderlich, falls enthalten)

Geben Sie die Dateien an, die in das Artefakt aufgenommen werden sollen. Sie müssen angeben "`cdk.out/**/*`", dass die synthetisierte AWS CloudFormation Vorlage Ihrer AWS CDK App eingeschlossen werden soll.

### Note

`cdk.out` ist das Standardverzeichnis, in dem synthetisierte Dateien gespeichert werden. Wenn Sie ein anderes Ausgabeverzeichnis als `cdk.out` in Ihrer `cdk.json` Datei angegeben haben, geben Sie dieses Verzeichnis hier statt `andck.out`.

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Artefakte/Artefakte hinzufügen/Von build erzeugte Dateien

## Environment

(*CDKDeploy*/Environment)

(Erforderlich)

Geben Sie die Umgebung an, die für die Aktion verwendet werden soll. CodeCatalyst

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ / „Umgebung/Konto/Rolle“ / Umgebung

## Name

(*CDKDeploy*/Environment/Name)

([Environment](#)Erforderlich, falls enthalten)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ / „Umgebung/Konto/Rolle“ / Umgebung

## Connections

(*CDKDeploy*/Environment/Connections)

([Environment](#)Erforderlich, falls enthalten)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können unter maximal eine Kontoverbindung angebenEnvironment.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen darüber, wie Sie eine Kontoverbindung mit Ihrer Umgebung verknüpfen, finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ / „Umgebung/Konto/Rolle“ / Kontoverbindung AWS

## Name

(*CDKDeploy*/Environment/Connections/Name)

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ / „Umgebung/Konto/Rolle“ / Kontoverbindung AWS

## Role

(*CDKDeploy*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die Bereitstellungsaktion für den Zugriff und die Bereitstellung des Anwendungsstapels verwendet AWS CDK . AWS AWS CDK Stellen Sie sicher, dass diese Rolle Folgendes umfasst:

- Die folgende Berechtigungsrichtlinie:

### Warning

Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie angegebenen. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
  "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::aws-account:role/cdk-*"
    }
  ]
}
```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Stellen Sie sicher, dass diese Rolle zu Ihrer Kontoverbindung hinzugefügt wird. Weitere Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

#### Note

Wenn Sie möchten, können Sie hier den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle angeben. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfangreiche Berechtigungen verfügt, was ein Sicherheitsrisiko darstellen kann. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger wichtig ist.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ / „Umgebung/Konto/Rolle“/

Configuration

(*CDKDeploy*/Configuration)

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“

**StackName**

(*CDKDeploy*/Configuration/StackName)


(Erforderlich)

Der Name Ihres AWS CDK App-Stacks, wie er in der Einstiegspunktdatei im Verzeichnis Ihrer AWS CDK App erscheint. *bin Das folgende Beispiel zeigt den Inhalt einer TypeScript Einstiegspunktdatei, wobei der Stackname rot kursiv hervorgehoben ist.* Wenn Ihre Einstiegspunktdatei in einer anderen Sprache ist, sieht sie ähnlich aus.

```
import * as cdk from 'aws-cdk-lib';
import { CdkWorkshopTypescriptStack } from '../lib/cdk_workshop_typescript-stack';

const app = new cdk.App();
new CdkWorkshopTypescriptStack(app, 'CdkWorkshopTypescriptStack');
```

Sie können nur einen Stapel angeben.

 **Tip**

Wenn Sie mehrere Stapel haben, können Sie einen übergeordneten Stapel mit verschachtelten Stacks erstellen. In dieser Aktion können Sie dann den übergeordneten Stapel angeben, um alle Stapel bereitzustellen.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/Name des Stacks

**Region**

(*CDKDeploy*/Configuration/Region)

(Erforderlich)

Geben Sie an, AWS-Region in welchem Bereich der AWS CDK Anwendungsstapel bereitgestellt werden soll. Eine Liste der Regionscodes finden Sie unter [Regionale Endpunkte](#).



Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration/Region“

## Tags

(*CDKDeploy*/Configuration/Tags)

(Optional)

Geben Sie die Tags an, die Sie auf die AWS Ressourcen im AWS CDK Anwendungsstapel anwenden möchten. Tags werden sowohl auf den Stack selbst als auch auf einzelne Ressourcen im Stack angewendet. Weitere Informationen zum Taggen finden Sie unter [Tagging](#) im AWS Cloud Development Kit (AWS CDK) Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ / „Erweitert“ — optional/ „Tags“

## Context

(*CDKDeploy*/Configuration/Context)

(Optional)

Geben Sie Kontexte in Form von Schlüssel-Wert-Paaren an, die dem Anwendungsstapel zugeordnet werden sollen. AWS CDK Weitere Informationen zu Kontexten finden Sie unter [Runtime-Kontexte](#) im AWS Cloud Development Kit (AWS CDK) Developer Guide.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“ /Erweitert — optional/ Kontext

## CdkCliVersion

(*CDKDeploy*/Configuration/CdkCliVersion)

(Optional)

Diese Eigenschaft ist in Version 1.0.13 oder höher der AWS CDK Bereitstellungsaktion und Version 1.0.8 oder höher der Bootstrap-Aktion verfügbar.AWS CDK

Geben Sie eines der folgenden Elemente an:

- Die Vollversion der AWS Cloud Development Kit (AWS CDK) Befehlszeilenschnittstelle (CLI) (auch AWS CDK Toolkit genannt), die diese Aktion verwenden soll. Beispiel: 2.102.1. Erwägen Sie die Angabe einer Vollversion, um Konsistenz und Stabilität beim Erstellen und Bereitstellen Ihrer Anwendung zu gewährleisten.

Oder

- `latest`. Erwägen Sie `latest` die Angabe, um die neuesten Funktionen und Korrekturen der CDK-CLI nutzen zu können.

Die Aktion lädt die angegebene Version (oder die neueste Version) der AWS CDK CLI auf das CodeCatalyst [Build-Image](#) herunter und verwendet dann diese Version, um die Befehle auszuführen, die für die Bereitstellung Ihrer CDK-Anwendung oder das Bootstrap Ihrer AWS Umgebung erforderlich sind.

Eine Liste der unterstützten CDK-CLI-Versionen, die Sie verwenden können, finden Sie unter [AWS CDK Versionen](#).

Wenn Sie diese Eigenschaft weglassen, verwendet die Aktion eine AWS CDK Standard-CLI-Version, die in einem der folgenden Themen beschrieben wird:

- [CDK-CLI-Versionen, die von der Aktion „AWS CDK Deploy“ verwendet werden](#)
- [CDK-CLI-Versionen, die von der Aktion „AWS CDK Bootstrap“ verwendet werden](#)

Entsprechende Benutzeroberfläche: Registerkarte „AWS CDK Konfiguration/CLI-Version“

## CdkRootPath

(*CDKDeploy*/Configuration/CdkRootPath)

(Optional)

Der Pfad zu dem Verzeichnis, das die `cdk.json` Datei Ihres AWS CDK Projekts enthält. Die AWS CDK Bereitstellungsaktion wird von diesem Ordner aus ausgeführt, und alle durch die Aktion erstellten Ausgaben werden diesem Verzeichnis hinzugefügt. Falls nicht angegeben, geht die AWS CDK Bereitstellungsaktion davon aus, dass sich die `cdk.json` Datei im Stammverzeichnis Ihres AWS CDK Projekts befindet.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/Verzeichnis, in dem sich die Datei `cdk.json` befindet

## CfnOutputVariables

(*CDKDeploy*/Configuration/CfnOutputVariables)

(Optional)

Geben Sie an, welche CfnOutput Konstrukte in Ihrem AWS CDK Anwendungscode Sie als Workflow-Ausgabevariablen verfügbar machen möchten. Sie können dann in nachfolgenden Aktionen in Ihrem Workflow auf die Workflow-Ausgabevariablen verweisen. Weitere Informationen zu Variablen finden Sie CodeCatalyst unter [Arbeiten mit Variablen](#).

Wenn Ihr AWS CDK Anwendungscode beispielsweise so aussieht:

```
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, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'bucketName', {
      value: bucket.bucketName,
      description: 'The name of the s3 bucket',
      exportName: 'myBucket',
    });
    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',
    });
    ...
  }
}
```

... und deine CfnOutputVariables Immobilie sieht so aus:

Configuration:

```
...
CfnOutputVariables: ["bucketName","tableName"]
```

... dann generiert die Aktion die folgenden Workflow-Ausgabevariablen:

| Schlüssel  | Wert              |
|------------|-------------------|
| bucketName | bucket.bucketName |
| tableName  | table.tableName   |

Sie können dann in nachfolgenden Aktionen auf die `tableName` Variablen `bucketName` und `tableName` verweisen. Informationen zum Verweisen auf Workflow-Ausgabevariablen in nachfolgenden Aktionen finden Sie unter [Verweisen auf eine vordefinierte Variable](#).

Wenn Sie in der `CfnOutputVariables` Eigenschaft keine `CfnOutput` Konstrukte angeben, macht die Aktion die ersten vier (oder weniger) gefundenen CloudFormation Ausgabevariablen als Workflow-Ausgabevariablen verfügbar. Weitere Informationen finden Sie unter [AWS CDK Aktionsvariablen „bereitstellen“](#).

#### Tip

Um eine Liste aller von der Aktion erzeugten CloudFormation Ausgabevariablen zu erhalten, führen Sie den Workflow, der die AWS CDK Bereitstellungsaktion enthält, einmal aus und schauen Sie dann auf der Registerkarte Protokolle der Aktion nach. Die Protokolle enthalten eine Liste aller mit Ihrer AWS CDK App verknüpften CloudFormation Ausgabevariablen. Sobald Sie wissen, was alle CloudFormation Variablen sind, können Sie mithilfe der `CfnOutputVariables` Eigenschaft angeben, welche Sie in Workflow-Ausgabevariablen konvertieren möchten.

Weitere Informationen zu AWS CloudFormation Ausgabevariablen finden Sie in der Dokumentation zum `CfnOutput` Konstrukt, die in der AWS Cloud Development Kit (AWS CDK) API-Referenz unter [class CfnOutput \(construct\)](#) verfügbar ist.

Entsprechende Benutzeroberfläche: Registerkarte „AWS CloudFormation Konfiguration/Ausgabevariablen“

CloudAssemblyRootPath

(*CDKDeploy*/Configuration/CloudAssemblyRootPath)

## (Optional)

Wenn Sie den Stack Ihrer AWS CDK App bereits zu einer Cloud-Assembly zusammengefasst haben (mithilfe der `cdk synth` Operation), geben Sie den Stammpfad des Cloud-Assembly-Verzeichnisses (`cdk.out`) an. Die AWS CloudFormation Vorlage, die sich im angegebenen Cloud-Assembly-Verzeichnis befindet, wird von der AWS CDK Bereitstellungsaktion AWS-Konto mithilfe des `cdk deploy --app` Befehls in Ihnen bereitgestellt. Wenn die `--app` Option vorhanden ist, findet der `cdk synth` Vorgang nicht statt.

Wenn Sie kein Cloud-Assembly-Verzeichnis angeben, führt die AWS CDK Bereitstellungsaktion den `cdk deploy` Befehl ohne die `--app` Option aus. Ohne die `--app` Option wird durch den `cdk deploy` Vorgang Ihre AWS CDK App sowohl synthetisiert (`cdk synth`) als auch in Ihrem AWS-Konto bereitgestellt.

Warum sollte ich eine bestehende, synthetisierte Cloud-Assembly angeben, wenn die Aktion „AWS CDK Deploy“ die Synthese zur Laufzeit durchführen kann?

Möglicherweise möchten Sie eine vorhandene, synthetisierte Cloud-Assembly spezifizieren, um:

- Stellen Sie sicher, dass bei jeder Ausführung der Aktion „AWS CDK Bereitstellen“ genau derselbe Satz von Ressourcen bereitgestellt wird

Wenn Sie keine Cloud-Assembly angeben, ist es möglich, dass die Aktion „AWS CDK Bereitstellen“ unterschiedliche Dateien synthetisiert und bereitstellt, je nachdem, wann sie ausgeführt wird. Die AWS CDK Bereitstellungsaktion könnte beispielsweise eine Cloud-Assembly mit einem Satz von Abhängigkeiten während einer Testphase und einem anderen Satz von Abhängigkeiten während einer Produktionsphase synthetisieren (wenn sich diese Abhängigkeiten zwischen den Phasen geändert haben). Um eine exakte Parität zwischen dem, was getestet wurde, und dem, was bereitgestellt wird, zu gewährleisten, empfehlen wir, einmal zu synthetisieren und dann das Feld Pfad zum Cloud-Assembly-Verzeichnis (visueller Editor) oder die `CloudAssemblyRootPath` Eigenschaft (YAML-Editor) zu verwenden, um die bereits synthetisierte Cloud-Assembly anzugeben.

- Verwenden Sie in der App nicht standardmäßige Paketmanager und Tools AWS CDK

Während eines `synth` Vorgangs versucht die AWS CDK Bereitstellungsaktion, Ihre App mithilfe von Standardtools wie `npm` oder `pip` auszuführen. Wenn die Aktion Ihre App mit diesen Tools nicht erfolgreich ausführen kann, findet die Synthese nicht statt und die Aktion schlägt fehl. Um dieses Problem zu umgehen, können Sie die genauen Befehle, die für die erfolgreiche Ausführung Ihrer App erforderlich sind, in der AWS CDK `cdk.json` App-Datei angeben und Ihre App dann mit einer

Methode synthetisieren, die nicht die AWS CDK Bereitstellungsaktion beinhaltet. Nachdem die Cloud-Assembly generiert wurde, können Sie sie im Feld Pfad zum Cloud-Assembly-Verzeichnis (visueller Editor) oder `CloudAssemblyRootPath` Eigenschaft (YAML-Editor) der AWS CDK Bereitstellungsaktion angeben.

Informationen zur Konfiguration der `cdk.json` Datei, sodass sie Befehle für die Installation und Ausführung Ihrer AWS CDK App enthält, finden Sie unter [Angeben des App-Befehls](#).

Informationen zu den `cdk synth` Befehlen `cdk deploy` und sowie zu den `--app` Optionen finden Sie unter [Stacks bereitstellen](#), [Stacks synthetisieren](#) und [Synthese überspringen](#) im Entwicklerhandbuch.AWS Cloud Development Kit (AWS CDK)

Informationen zu Cloud-Assemblies finden Sie unter [Cloud Assembly in der API-Referenz](#).AWS Cloud Development Kit (AWS CDK)

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/Pfad zum Cloud-Assembly-Verzeichnis

## „AWS Lambda Aufruf“-Aktionsreferenz

Im Folgenden finden Sie die YAML-Referenz der Aktionsdefinition für die AWS Lambda Aufrufaktion. Informationen zur Verwendung dieser Aktion finden Sie unter [Hinzufügen der Aktion „AWS LambdaAufrufen“](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das -Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.  
# See Immobilien auf höchstem Niveau 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

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: `Lambda_Invoke_Action_Workflow_nn`.

Entsprechende Benutzeroberfläche: Registerkarte/Aktionsname der Konfiguration

## Identifizier

(*LambdaInvoke*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: `aws/lambda-invoke@v1`.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/LambdaInvoke\_nn/aws/lambda-invoke@v1-Bezeichnung

## DependsOn

(*LambdaInvoke*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

(*LambdaInvoke*/Compute)



(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

(*LambdaInvoke*/Compute/Type)

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsausführungen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Datenverarbeitungstyp Konfiguration

## Fleet

(*LambdaInvoke*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen `Fleet` wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Datenverarbeitungsflotte

## Timeout

(*LambdaInvoke*/Timeout)

(Erforderlich)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Inputs

(*LambdaInvoke*/Inputs)

(Erforderlich)

Der `Inputs` Abschnitt definiert die Daten, die die AWS Lambda Aufrufaktion während einer Workflow-Ausführung benötigt.

### Note

Pro AWS Lambda Aufrufaktion ist nur eine Eingabe (entweder eine Quelle oder ein Artefakt) zulässig. Variablen werden nicht auf diese Summe angerechnet.

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

## Sources

(*LambdaInvoke*/Inputs/Sources)

(Erforderlich, wenn angegeben [RequestPayloadFile](#) ist)

Wenn Sie eine JSON-Anforderungsnutzlastdatei an die AWS Lambda Aufrufaktion übergeben möchten und diese Nutzlastdatei in einem Quell-Repository gespeichert ist, geben Sie die Bezeichnung dieses Quell-Repositorys an. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`.

Wenn Ihre Anforderungsnutzlastdatei nicht in einem Quell-Repository enthalten ist, muss sie sich in einem Artefakt befinden, das von einer anderen Aktion generiert wird.

Weitere Informationen zur Nutzlastdatei finden Sie unter [RequestPayloadFile](#).

#### Note

Anstatt eine Nutzlastdatei anzugeben, können Sie den JSON-Code der Nutzlast mithilfe der `-RequestPayloadEigenschaft` direkt zur `-Aktion` hinzufügen. Weitere Informationen finden Sie unter [RequestPayload](#).

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

(*LambdaInvoke*/Inputs/Artifacts)

(Erforderlich, wenn angegeben [RequestPayloadFile](#) ist)

Wenn Sie eine JSON-Anforderungsnutzlastdatei an die AWS Lambda Aufrufaktion übergeben möchten und diese Nutzlastdatei in einem [Ausgabeartefakt](#) aus einer vorherigen Aktion enthalten ist, geben Sie dieses Artefakt hier an.

Weitere Informationen zur Nutzlastdatei finden Sie unter [RequestPayloadFile](#).

#### Note

Anstatt eine Nutzlastdatei anzugeben, können Sie den JSON-Code der Nutzlast mithilfe der `-RequestPayloadEigenschaft` direkt zur `-Aktion` hinzufügen. Weitere Informationen finden Sie unter [RequestPayload](#).

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte der Konfiguration – optional

## Variables - input

(*LambdaInvoke*/Inputs/Variables)

(Optional)

Geben Sie eine Folge von Name/Wert-Paaren an, die die Eingabevariablen definieren, die Sie der Aktion zur Verfügung stellen möchten. Variablennamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Variablennamen zu aktivieren.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Variablen Eingaben – optional

## Environment

(*LambdaInvoke*/Environment)

(Erforderlich)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

## Name

(*LambdaInvoke*/Environment/Name)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/„Umgebung/Verbindung/  
Rolle“/Umgebung

## Connections

(*LambdaInvoke*/Environment/Connections)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können unter maximal eine Kontoverbindung angeben `Environment`.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/ „Umgebung/Verbindung/  
Rolle“/Verbindung

## Name

(*LambdaInvoke*/Environment/Connections/Name)

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/„Umgebung/Verbindung/  
Rolle“/Verbindung

## Role

(*LambdaInvoke*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die AWS Lambda Aufrufaktion verwendet, um auf Ihre Lambda-Funktion zuzugreifen AWS und diese aufzurufen. Stellen Sie sicher, dass diese Rolle Folgendes beinhaltet:

- Die folgende Berechtigungsrichtlinie:

**⚠ Warning**


Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie aufgeführten. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:aws-account:function:function-name"
    }
  ]
}
```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Stellen Sie sicher, dass diese Rolle mit Ihrer Kontoverbindung verknüpft ist. Weitere Informationen zum Zuordnen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

 Note

Sie können den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/ „Umgebung/Verbindung/Rolle“/Rolle

## Configuration

(*LambdaInvoke*/Configuration)

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## Function

(*LambdaInvoke*/Configuration/Function)

(Erforderlich)

Geben Sie die AWS Lambda Funktion an, die diese Aktion aufruft. Sie können den Namen der Funktion oder ihren Amazon-Ressourcennamen (ARN) angeben. Sie finden den Namen oder ARN in der Lambda-Konsole.

**Note**

Das AWS Konto, in dem sich die Lambda-Funktion befindet, kann sich von dem unter angegebenen Konto unterscheiden: `Connections` .

Entsprechende Benutzeroberfläche: Registerkarte/Funktion „Konfiguration“

**AWSRegion**

(*LambdaInvoke*/Configuration/AWSRegion)

(Erforderlich)

Geben Sie die AWS Region an, in der sich Ihre AWS Lambda Funktion befindet. Eine Liste der Regionscodes finden Sie unter [Regionale Endpunkte](#) im Allgemeine AWS-Referenz.

Entsprechende Benutzeroberfläche: Konfigurations-Registerkarte/Ziel-Bucket – optional

**RequestPayload**

(*LambdaInvoke*/Configuration/RequestPayload)

(Optional)

Wenn Sie eine Anforderungsnutzlast an die AWS Lambda Aufrufaktion übergeben möchten, geben Sie die Anforderungsnutzlast hier im JSON-Format an.

Beispiel für eine Anforderungsnutzlast:

```
'{ "key": "value" }'
```

Wenn Sie keine Anforderungsnutzlast an Ihre Lambda-Funktion übergeben möchten, lassen Sie diese Eigenschaft weg.

**Note**

Sie können `RequestPayload` oder `RequestPayloadFile` angeben, aber nicht beides.

Weitere Informationen zur Anforderungsnutzlast finden Sie im Thema [Aufrufen](#) in der APIAWS Lambda -Referenz zu .



Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Nutzlast anfordern – optional

## RequestPayloadFile

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Optional)

Wenn Sie eine Anforderungsnutzlast an die AWS Lambda Aufrufaktion übergeben möchten, geben Sie hier den Pfad zu dieser Anforderungsnutzlastdatei an. Die Datei muss im JSON-Format vorliegen.

Die Anforderungsnutzlastdatei kann sich in einem Quell-Repository oder einem Artefakt aus einer vorherigen Aktion befinden. Der Dateipfad ist relativ zum Quell-Repository oder Artefaktstamm.

Wenn Sie keine Anforderungsnutzlast an Ihre Lambda-Funktion übergeben möchten, lassen Sie diese Eigenschaft weg.

### Note

Sie können `RequestPayload` oder `RequestPayloadFile` angeben, aber nicht beides.

Weitere Informationen zur Anforderungsnutzlastdatei finden Sie im Thema [Aufrufen](#) in der APIAWS Lambda -Referenz zu .

Entsprechende Benutzeroberfläche: Konfigurationsregisterkarte/Nutzlastdatei anfordern – optional

## ContinueOnError

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Optional)

Geben Sie an, ob Sie die AWS Lambda Aufrufaktion auch dann als erfolgreich markieren möchten, wenn die aufgerufene AWS Lambda Funktion fehlschlägt. Erwägen Sie, diese Eigenschaft auf `true` zu setzen, damit nachfolgende Aktionen in Ihrem Workflow trotz des Lambda-Fehlers gestartet werden können.

Standardmäßig schlägt die Aktion fehl, wenn die Lambda-Funktion im visuellen Editor oder `false` im YAML-Editor fehlschlägt („off“).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Bei Fehler fortfahren

## LogType

(*LambdaInvoke*/Configuration/LogType)

(Optional)

Geben Sie an, ob Sie Fehlerprotokolle nach dem Aufruf in die Antwort der Lambda-Funktion aufnehmen möchten. Sie können diese Protokolle auf der Registerkarte Protokolle der Lambda-Aufrufaktion in der - CodeCatalyst Konsole anzeigen. Die möglichen Werte sind:

- `Tail` – Rückgabeprotokolle
- `None` – gibt keine Protokolle zurück

Der Standardwert ist `Tail` .

Weitere Informationen zum Protokolltyp finden Sie unter dem Thema [Aufrufen](#) in der APIAWS Lambda -Referenz zu .

Weitere Informationen zum Anzeigen von Protokollen finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).

Entsprechende Benutzeroberfläche: Registerkarte/Protokolltyp Konfiguration

## ResponseFilters

(*LambdaInvoke*/Configuration/ResponseFilters)

(Optional)

Geben Sie an, welche Schlüssel in der Lambda-Antwortnutzlast Sie in Ausgabevariablen konvertieren möchten. Anschließend können Sie in nachfolgenden Aktionen in Ihrem Workflow auf die Ausgabevariablen verweisen. Weitere Informationen zu Variablen in CodeCatalystfinden Sie unter [Arbeiten mit Variablen](#).

Wenn Ihre Antwortnutzlast beispielsweise wie folgt aussieht:

```
responsePayload = {  
  "name": "Saanvi",
```

```

"location": "Seattle",
"department": {
  "company": "Amazon",
  "team": "AWS"
}
}

```

... und Ihre Antwortfilter sehen wie folgt aus:

```

Configuration:
...
ResponseFilters: '{"name": ".name", "company": ".department.company"}'

```

... dann generiert die Aktion die folgenden Ausgabevariablen:

| Schlüssel | Wert   |
|-----------|--------|
| Name      | Saanvi |
| company   | Amazon |

Sie können dann in nachfolgenden Aktionen auf die `company` Variablen `name` und `company` verweisen.

Wenn Sie keine Schlüssel in `answerResponseFilters` angeben, konvertiert die Aktion jeden Schlüssel der obersten Ebene in der Lambda-Antwort in eine Ausgabevariable. Weitere Informationen finden Sie unter [AWS Lambda Aktionsvariablen „aufrufen“](#).

Erwägen Sie die Verwendung von Antwortfiltern, um die generierten Ausgabevariablen auf diejenigen zu beschränken, die Sie tatsächlich verwenden möchten.

Entsprechende Benutzeroberfläche: Konfigurations-Registerkarte/Antwortfilter – optional

## Outputs

(*LambdaInvoke*/Outputs)

(Optional)

Definiert die Daten, die von der Aktion während einer Workflow-Ausführung ausgegeben werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“

## Artifacts

(*LambdaInvoke*/Outputs/Artifacts)

(Optional)

Geben Sie die Artefakte an, die von der Aktion generiert wurden. Sie können diese Artefakte als Eingabe in anderen Aktionen referenzieren.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Artifacts/Build-Artefaktnamen

## Name

(*LambdaInvoke*/Outputs/Artifacts/Name)

(Optional)

Geben Sie den Namen des Artefakts an, das die Lambda-Antwortnutzlast enthält, die von der Lambda-Funktion zurückgegeben wird. Der Standardwert ist `lambda_artifacts`. Wenn Sie kein Artefakt angeben, kann die Lambda-Antwortnutzlast in den Protokollen der Aktion angezeigt werden, die auf der Registerkarte Protokolle für die Aktion in der CodeCatalyst Konsole verfügbar sind.

Weitere Informationen zum Anzeigen von Protokollen finden Sie unter [Anzeigen des Status und der Details der Workflow-Ausführung](#).

Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Artifacts/Build-Artefaktnamen

## Files

(*LambdaInvoke*/Outputs/Artifacts/Files)

(Optional)

Geben Sie die Dateien an, die in das Artefakt aufgenommen werden sollen. Sie müssen angeben, `lambda-response.json` damit die Lambda-Antwortnutzlastdatei enthalten ist.

Entsprechende Benutzeroberfläche: Registerkarte „Outputs“/„Artifacts“/„Files“, die vom Build erstellt wurden

## Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“

Im Folgenden finden Sie die YAML-Referenz für die Aktionsdefinition für die Aktion AWS CloudFormationStack bereitstellen. Informationen zur Verwendung dieser Aktion finden Sie unter [Aktion „AWS CloudFormation Stack bereitstellen“ hinzufügen](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das -Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau 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

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: DeployCloudFormationStack\_nn.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Anzeigename der Aktion

### Identifizier

(*DeployCloudFormationStack/Identifizier*)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: aws/cfn-deploy@v1.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/DeployCloudFormationStack\_nn/aws/cfn-deploy@v1-Bezeichnung

## DependsOn

*(DeployCloudFormationStack/DependsOn)*

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

*(DeployCloudFormationStack/Compute)*

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

*(DeployCloudFormationStack/Compute/Type)*

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsausführungen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungstyp

## Fleet

*(DeployCloudFormationStack/Compute/Fleet)*

(Optional)

Geben Sie die Maschine oder Flotte an, die Ihren Workflow oder Ihre Workflow-Aktionen ausführen wird. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen Fleet wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungsflotte

## Timeout

*(DeployCloudFormationStack/Timeout)*

(Optional)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout in Minuten – optional

## Environment

*(DeployCloudFormationStack/Environment)*



(Erforderlich)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

Name

*(DeployCloudFormationStack/Environment/Name)*

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

Connections

*(DeployCloudFormationStack/Environment/Connections)*

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können maximal eine Kontoverbindung unter angebenEnvironment.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

Name

*(DeployCloudFormationStack/Environment/Connections/Name)*

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Role

(*DeployCloudFormationStack*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die Aktion AWS CloudFormationStack bereitstellen für den Zugriff auf AWS und den AWS CloudFormation Service verwendet.

Stellen Sie sicher, dass diese Rolle die folgenden Richtlinien enthält:

- Die folgende Berechtigungsrichtlinie:

### Warning

Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie aufgeführten. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

}

**Note**

Wenn die Rolle zum ersten Mal verwendet wird, verwenden Sie den folgenden Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach ihrer Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Stellen Sie sicher, dass diese Rolle der Kontoverbindung in Ihrem Bereich hinzugefügt wird. Weitere Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

**Note**

Sie können den Namen der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die

CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Rolle

## Inputs

(*DeployCloudFormationStack/Inputs*)

(Optional)

Der Inputs Abschnitt definiert die Daten, die der während einer Workflow-Ausführung DeployCloudFormationStack benötigt.

### Note

Pro Deploy-AWS CloudFormationStack-Aktion sind maximal vier Eingaben (eine Quelle und drei Artefakte) zulässig.

Wenn Sie auf Dateien verweisen müssen, die sich in verschiedenen Eingaben befinden (z. B. eine Quelle und ein Artefakt), ist die Quelleingabe die primäre Eingabe und das Artefakt ist die sekundäre Eingabe. Verweise auf Dateien in sekundären Eingaben verwenden ein spezielles Präfix, um sie vom primären zu trennen. Details hierzu finden Sie unter [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

## Sources

(*DeployCloudFormationStack/Inputs/Sources*)

(Erforderlich, wenn Ihre - CloudFormation oder -AWS SAMVorlage in einem Quell-Repository gespeichert ist)

Wenn Ihre - CloudFormation oder -AWS SAMVorlage in einem Quell-Repository gespeichert ist, geben Sie die Bezeichnung dieses Quell-Repositorys an. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`.

Wenn Ihre - CloudFormation oder -AWS SAMVorlage nicht in einem Quell-Repository enthalten ist, muss sie sich in einem Artefakt befinden, das durch eine andere Aktion generiert wurde, oder in einem Amazon S3-Bucket.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

*(DeployCloudFormationStack/Inputs/Artifacts)*

(Erforderlich, wenn Ihre - CloudFormation oder -AWS SAMVorlage in einem [Ausgabeartefakt](#) aus einer vorherigen Aktion gespeichert ist)

Wenn die - CloudFormation oder -AWS SAMVorlage, die Sie bereitstellen möchten, in einem Artefakt enthalten ist, das von einer vorherigen Aktion generiert wurde, geben Sie dieses Artefakt hier an. Wenn Ihre CloudFormation Vorlage nicht in einem Artefakt enthalten ist, muss sie sich in Ihrem Quell-Repository oder in einem Amazon S3-Bucket befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte der Konfiguration – optional

## Configuration

*(DeployCloudFormationStack/Configuration)*

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

### name

*(DeployCloudFormationStack/Configuration/name)*

(Erforderlich)

Geben Sie einen Namen für den CloudFormation Stack an, den die Aktion AWS CloudFormationStack bereitstellen erstellt oder aktualisiert.

Entsprechende Benutzeroberfläche: Registerkarte/Stack-Name der Konfiguration

region

*(DeployCloudFormationStack/Configuration/region)*

(Erforderlich)

Geben Sie die an, AWS-Region in der der Stack bereitgestellt werden soll. Eine Liste der Regionscodes finden Sie unter [Regionale Endpunkte](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Stack-Region

template

*(DeployCloudFormationStack/Configuration/template)*

(Erforderlich)


Geben Sie den Namen und den Pfad zu Ihrer - CloudFormation oder -AWS SAMVorlagendatei an. Die Vorlage kann im JSON- oder YAML-Format vorliegen und sich in einem Quell-Repository, einem Artefakt aus einer vorherigen Aktion oder einem Amazon S3-Bucket befinden. Wenn sich die Vorlagendatei in einem Quell-Repository oder Artefakt befindet, ist der Pfad relativ zum Quell- oder Artefaktstamm. Wenn sich die Vorlage in einem Amazon S3-Bucket befindet, ist der Pfad der Wert der Objekt-URL der Vorlage.

Beispiele:

`./MyFolder/MyTemplate.json`

`MyFolder/MyTemplate.yml`

`https://MyBucket.s3.us-west-2.amazonaws.com/MyTemplate.yml`

 Note

Möglicherweise müssen Sie dem Dateipfad der Vorlage ein Präfix hinzufügen, um anzugeben, in welchem Artefakt oder welcher Quelle sie gefunden werden soll. Weitere Informationen finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

Entsprechende Benutzeroberfläche: Registerkarte/Vorlage „Konfiguration“

## role-arn

(*DeployCloudFormationStack*/Configuration/role-arn)

(Erforderlich)

Geben Sie den Amazon-Ressourcennamen (ARN) der Stack-Rolle an. CloudFormation verwendet diese Rolle, um auf Ressourcen in Ihrem Stack zuzugreifen und diese zu ändern. Zum Beispiel: `arn:aws:iam::123456789012:role/StackRole`.

Stellen Sie sicher, dass die Stack-Rolle Folgendes enthält:

- Eine oder mehrere Berechtigungsrichtlinien. Die Richtlinien hängen von den Ressourcen ab, die Sie in Ihrem Stack haben. Wenn Ihr Stack beispielsweise eine -AWS LambdaFunktion enthält, müssen Sie Berechtigungen hinzufügen, die Zugriff auf Lambda gewähren. Wenn Sie das unter beschriebene Tutorial befolgt haben [Tutorial: Bereitstellen einer serverlosen Anwendung mit AWS CloudFormation](#), enthält es ein Verfahren mit dem Titel [Um eine Stack-Rolle zu erstellen](#), das die Berechtigungen auflistet, die die Stack-Rolle benötigt, wenn Sie einen typischen Serverless-Anwendungs-Stack bereitstellen.

### Warning

Beschränken Sie die Berechtigungen auf diejenigen, die der CloudFormation Service für den Zugriff auf Ressourcen in Ihrem Stack benötigt. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

- Die folgende Vertrauensrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudformation.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Ordnen Sie diese Rolle optional Ihrer Kontoverbindung zu. Weitere Informationen zum Zuordnen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#). Wenn Sie die Stack-Rolle nicht mit der Kontoverbindung verknüpfen, wird die Stack-Rolle nicht in der Dropdown-Liste Stack-Rolle im visuellen Editor angezeigt. Der Rollen-ARN kann jedoch weiterhin im `role-arn` Feld mit dem YAML-Editor angegeben werden.

 Note

Sie können den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Entsprechende Benutzeroberfläche: Registerkarte/Stack-Rolle Konfiguration – optional capabilities

(*DeployCloudFormationStack*/Configuration/capabilities)

(Erforderlich)

Geben Sie eine Liste der IAM-Funktionen an, die erforderlich sind, damit bestimmte AWS CloudFormation Stacks erstellen kann. In den meisten Fällen können Sie capabilities mit dem Standardwert belassen `CAPABILITY_IAM, CAPABILITY_NAMED_IAM, CAPABILITY_AUTO_EXPAND`.

Wenn Sie `##[error] requires capabilities: [capability-name]` in den Protokollen Ihrer Deploy-AWS CloudFormationStack-Aktion sehen, finden Sie unter [Wie behebe ich Fehler bei den IAM-Fähigkeiten?](#) Informationen zur Behebung des Problems.

Weitere Informationen zu IAM-Funktionen finden Sie unter [Bestätigung von IAM-Ressourcen in -AWS CloudFormationVorlagen](#) im IAM-Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert/Funktionen



## parameter-overrides

(*DeployCloudFormationStack*/Configuration/parameter-overrides)

(Optional)

Geben Sie Parameter in Ihrer - AWS CloudFormation oder -AWS SAMVorlage an, die keine Standardwerte haben oder für die Sie Nicht-Standardwerte angeben möchten. Weitere Informationen zu Parametern finden Sie unter [Parameter](#) im AWS CloudFormation-Benutzerhandbuch.

Die -parameter-overridesEigenschaft akzeptiert:

- Eine JSON-Datei, die die Parameter und Werte enthält.
- Eine durch Komma getrennte Liste von Parametern und Werten.

So geben Sie eine JSON-Datei an

1. Stellen Sie sicher, dass die JSON-Datei eine der folgenden Syntaxen verwendet:

```
{
  "Parameters": {
    "Param1": "Value1",
    "Param2": "Value2",
    ...
  }
}
```

Oder...

```
[
  {
    "ParameterKey": "Param1",
    "ParameterValue": "Value1"
  },
  ...
]
```

(Es gibt andere Syntaxen, aber sie werden von CodeCatalyst zum Zeitpunkt des Schreibens nicht unterstützt.) Weitere Informationen zum Angeben von CloudFormation Parametern in einer JSON-Datei finden Sie unter [Unterstützte JSON-Syntax](#) in der AWS CLI -Befehlsreferenz.

## 2. Geben Sie den Pfad zur JSON-Datei mit einem der folgenden Formate an:

- Wenn sich Ihre JSON-Datei in einem Ausgabeartefakt aus einer vorherigen Aktion befindet, verwenden Sie:

```
file:///artifacts/current-action-name/output-artifact-name/path-to-json-file
```

Einzelheiten finden Sie in Beispiel 1.

- Wenn sich Ihre JSON-Datei in Ihrem Quell-Repository befindet, verwenden Sie:

```
file:///sources/WorkflowSource/path-to-json-file
```

Einzelheiten finden Sie in Beispiel 2.

Beispiel 1 – Die JSON-Datei befindet sich in einem Ausgabeartefakt

```
##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
```

Beispiel 2 – Die JSON-Datei befindet sich in Ihrem Quell-Repository in einem Ordner mit dem Namen my/folder

```
##My workflow YAML
...
Actions:
  MyDeployCloudFormationStack:
```

```
Identifizier: aws/cfn-deploy@v1
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  parameter-overrides: file:///sources/WorkflowSource/my/folder/params.json
```

So verwenden Sie eine durch Komma getrennte Liste von Parametern

- Fügen Sie Parameter-Name-Wert-Paare in der `-parameter-overrides`Eigenschaft im folgenden Format hinzu:

*param-1=value-1,param-2=value-2*

Angenommen, es wird die folgende AWS CloudFormation Vorlage verwendet:

```
##My CloudFormation template

Description: My AWS 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:
  ...
```

... Sie können die `parameter-overrides` Eigenschaft wie folgt festlegen:

```
##My workflow YAML
...
Actions:
  ...
  DeployCloudFormationStack:
```

Identifizier: aws/cfn-deploy@v1

Configuration:

**parameter-overrides: InstanceType=t3.medium,UseVPC=true**

### Note

Sie können einen Parameternamen ohne einen entsprechenden Wert angeben, indem Sie `undefined` als Wert verwenden. Beispielsweise:

`parameter-overrides: MyParameter=undefined`

Dies hat zur Folge, dass während einer Stack-Aktualisierung den vorhandenen Parameterwert für den angegebenen Parameternamen CloudFormation verwendet.

Entsprechende Benutzeroberfläche:

- Konfigurations-Registerkarte/Erweitert/Parameter-Überschreibungen
- Konfigurations-Registerkarte/Erweitert/Parameterüberschreibungen/Überschreibungen mithilfe einer Datei angeben
- Registerkarte Konfiguration/Erweitert/Parameterüberschreibungen/Überschreibungen unter Verwendung eines Wertsatzes angeben

`no-execute-changeset`

*(DeployCloudFormationStack/Configuration/no-execute-changeset)*

(Optional)

Geben Sie CodeCatalyst an, ob Sie den CloudFormation Änderungssatz erstellen und dann anhalten möchten, bevor Sie ihn ausführen. Auf diese Weise haben Sie die Möglichkeit, den Änderungssatz in der CloudFormation Konsole zu überprüfen. Wenn Sie feststellen, dass der Änderungssatz gut aussieht, deaktivieren Sie diese Option und führen Sie den Workflow erneut aus, damit den Änderungssatz erstellen und ausführen CodeCatalyst kann, ohne ihn zu beenden. Standardmäßig wird der Änderungssatz erstellt und ausgeführt, ohne ihn anzuhalten. Weitere Informationen finden Sie unter dem AWS CloudFormation [Bereitstellungsparameter](#) in der AWS CLI -Befehlsreferenz. Weitere Informationen zum Anzeigen von Änderungssätzen finden Sie unter [Anzeigen eines Änderungssatzes](#) im AWS CloudFormation -Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert/Keine Ausführungsänderungssatz

fail-on-empty-changeset

(*DeployCloudFormationStack*/Configuration/fail-on-empty-changeset)

(Optional)

Geben Sie CodeCatalyst an, ob die Aktion AWS CloudFormationStack bereitstellen fehlschlagen soll, wenn der CloudFormation Änderungssatz leer ist. (Wenn ein Änderungssatz leer ist, bedeutet dies, dass während der letzten Bereitstellung keine Änderungen am Stack vorgenommen wurden.) Standardmäßig wird die Aktion fortgesetzt, wenn der Änderungssatz leer ist, und eine -UPDATE\_COMPLETE-Nachricht zurückgegeben, obwohl der Stack nicht aktualisiert wurde.

Weitere Informationen zu dieser Einstellung finden Sie im AWS CloudFormation [Bereitstellungsparameter](#) in der AWS CLI -Befehlsreferenz. Weitere Informationen zu Änderungssätzen finden Sie unter [Aktualisieren von Stacks mithilfe von Änderungssätzen](#) im AWS CloudFormation -Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert/Fehler bei leerem Änderungssatz

disable-rollback

(*DeployCloudFormationStack*/Configuration/disable-rollback)

(Optional)

Geben Sie CodeCatalyst an, ob Sie die Stack-Bereitstellung zurücksetzen möchten, falls sie fehlschlägt. Das Rollback setzt den Stack in den letzten bekannten stabilen Zustand zurück. Standardmäßig werden Rollbacks aktiviert. Weitere Informationen zu dieser Einstellung finden Sie im AWS CloudFormation [Bereitstellungsparameter](#) in der AWS CLI -Befehlsreferenz.

Weitere Informationen darüber, wie die Aktion AWS CloudFormationStack bereitstellen mit Rollbacks umgeht, finden Sie unter [Rollbacks konfigurieren](#).

Weitere Informationen zum Rollback eines Stacks finden Sie unter [Stack-Fehleroptionen](#) im AWS CloudFormation -Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Konfigurations-Tab/Erweitert/Deaktivieren des Rollbacks

## termination-protection

*(DeployCloudFormationStack/Configuration/termination-protection)*

(Optional)

Geben Sie an, ob der Deploy-AWS CloudFormationStack dem Stack, den er bereitstellt, einen Beendigungsschutz hinzufügen soll. Wenn ein Benutzer versucht, einen Stack mit aktiviertem Beendigungsschutz zu löschen, schlägt das Löschen fehl und der Stack – einschließlich seines Status – bleibt unverändert. Standardmäßig wird der Beendigungsschutz deaktiviert. Weitere Informationen finden Sie unter [Schützen eines Stacks vor dem Löschen](#) im AWS CloudFormation - Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Konfigurations-Tab/Erweitert/Beendigungsschutz

## timeout-in-minutes

*(DeployCloudFormationStack/Configuration/timeout-in-minutes)*

(Optional)

Geben Sie die Zeit in Minuten an, die zuweisen CloudFormation soll, bevor ein Timeout für Operationen zur Stack-Erstellung durchgeführt und der Stack-Status auf festgelegt wirdCREATE\_FAILED. Wenn CloudFormation innerhalb des zugeteilten Zeitraums nicht den gesamten Stack erstellen kann, schlägt die Stack-Erstellung aufgrund des Timeouts fehl und für den Stack wird ein Rollback durchgeführt.

Standardmäßig gibt es keinen Timeout für die Stack-Erstellung. Allerdings gibt es möglicherweise für einzelne Ressourcen einen Timeout, basierend auf der Art des von ihnen implementierten Services. Wenn beispielsweise eine einzelne Ressource in Ihrem Stack abläuft, läuft die Stack-Erstellung ebenfalls aus, auch wenn der von Ihnen angegebene Timeout für die Stack-Erstellung noch nicht erreicht wurde.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert/CloudFormationTimeout

## notification-arns

*(DeployCloudFormationStack/Configuration/notification-arns)*

(Optional)

Geben Sie den ARN eines Amazon SNS-Themas CodeCatalyst an, an das Sie Benachrichtigungen senden möchten. Beispiel: `arn:aws:sns:us-east-1:111222333:MyTopic` Wenn die Aktion `AWS CloudFormationStack bereitstellen` ausgeführt wird, CodeCatalyst koordiniert mit , CloudFormation um eine Benachrichtigung pro AWS CloudFormation Ereignis zu senden, das während des Stack-Erstellungs- oder Aktualisierungsprozesses auftritt. (Die Ereignisse sind auf der Registerkarte Ereignisse der AWS CloudFormation-Konsole für den Stack sichtbar.) Sie können bis zu fünf Themen angeben. Weitere Informationen finden Sie unter [Was ist Amazon SNS?](#).


Entsprechende Benutzeroberfläche: Konfigurations-Registerkarte/Erweitert/Benachrichtigungs-ARNs  
`monitor-alarm-arns`

*(DeployCloudFormationStack/Configuration/monitor-alarm-arns)*

(Optional)

Geben Sie den Amazon-Ressourcennamen (ARN) eines Amazon- CloudWatch Alarms an, der als Rollback-Auslöser verwendet werden soll. Beispiel:

`arn:aws:cloudwatch::123456789012:alarm/MyAlarm` Sie können maximal fünf Rollback-Auslöser haben.

 Note

Wenn Sie einen CloudWatch Alarm-ARN angeben, müssen Sie auch zusätzliche Berechtigungen konfigurieren, damit die Aktion auf zugreifen kann CloudWatch. Weitere Informationen finden Sie unter [Rollbacks konfigurieren](#).

Entsprechende Benutzeroberfläche: Konfigurations-Registerkarte/Erweitert/Überwachung von Alarm-ARNs

`monitor-timeout-in-minutes`

*(DeployCloudFormationStack/Configuration/monitor-timeout-in-minutes)*

(Optional)

Geben Sie eine Zeitspanne von 0 bis 180 Minuten an, in der die angegebenen Alarme CloudFormation überwacht. Die Überwachung beginnt, nachdem alle Stack-Ressourcen bereitgestellt wurden. Wenn der Alarm innerhalb der angegebenen Überwachungszeit auftritt, schlägt die Bereitstellung fehl und CloudFormation setzt den gesamten Stack-Vorgang zurück.

Standard: 0. überwacht CloudFormation nur Alarme, während die Stack-Ressourcen bereitgestellt werden, nicht danach.

Entsprechende Benutzeroberfläche: Konfigurations-Tab/Erweitert/Überwachungszeit

tags

`(DeployCloudFormationStack/Configuration/tags)`

(Optional)

Geben Sie Tags an, die an Ihren CloudFormation Stack angehängt werden sollen. Tags sind beliebige Schlüssel-Wert-Paare, mit denen Sie Ihren Stack für Zwecke wie die Kostenzuordnung identifizieren können. Weitere Informationen darüber, was Tags sind und wie sie verwendet werden können, finden Sie unter [Markieren Ihrer -Ressourcen](#) im Amazon EC2-Benutzerhandbuch für Linux-Instances. Weitere Informationen zum Markieren in CloudFormation finden Sie unter [Festlegen von AWS CloudFormationStack-Optionen](#) im AWS CloudFormation -Benutzerhandbuch.

Ein Schlüssel kann alphanumerische Zeichen oder Leerzeichen und bis zu 127 Zeichen enthalten. Ein Wert kann alphanumerische Zeichen oder Leerzeichen und bis zu 255 Zeichen lang sein.

Sie können bis zu 50 eindeutige Tags für jeden Stack hinzufügen.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert/Tags

## Referenz zur Aktion „Deploy to Amazon ECS“

Im Folgenden finden Sie die YAML-Referenz der Aktionsdefinition für die Aktion Deploy to Amazon ECS. Informationen zur Verwendung dieser Aktion finden Sie unter [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das -Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.  
# See Immobilien auf höchstem Niveau 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: DeployToECS
  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

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: `ECSDeployAction_nn`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Anzeigename der Aktion

## Identifizier

(*ECSDeployAction*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten.

Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: `aws/ecs-deploy@v1`.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/ECS DeployAction\_nn/aws/ecs-deploy@v1-  
Bezeichnung

## DependsOn

(*ECSDeployAction*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

(*ECSDeployAction*/Compute)

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

(*ECSDeployAction*/Compute/Type)

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsausführungen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungstyp

## Fleet

(*ECSDeployAction*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen Fleet wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungsflotte

## Timeout

(*ECSDeployAction*/Timeout)

(Optional)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Environment

(*ECSDeployAction*/Environment)

(Erforderlich)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

## Name

(*ECSDeployAction*/Environment/Name)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

## Connections

(*ECSDeployAction*/Environment/Connections)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können unter maximal eine Kontoverbindung angeben `Environment`.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Name

(*ECSDeployAction*/Environment/Connections/Name)

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Role

(*ECSDeployAction*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die Aktion Deploy to Amazon ECS für den Zugriff auf verwendet AWS. Stellen Sie sicher, dass diese Rolle die folgenden Richtlinien enthält:

- Die folgende Berechtigungsrichtlinie:

### Warning

Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie aufgeführten. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
```

```
"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

Wenn die Rolle zum ersten Mal verwendet wird, verwenden Sie den folgenden Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach ihrer Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"
```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

```
]
}
```

Stellen Sie sicher, dass diese Rolle zu Ihrer Kontoverbindung hinzugefügt wird. Weitere Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

#### Note

Sie können den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Rolle

## Inputs

(*ECSDeployAction*/Inputs)

(Optional)

Der Inputs Abschnitt definiert die Daten, die der während einer Workflow-Ausführung `ECSDeployAction` benötigt.

#### Note

Pro Aktion `Deploy to Amazon ECS` ist nur eine Eingabe (entweder eine Quelle oder ein Artefakt) zulässig.

Entsprechende Benutzeroberfläche: Registerkarte Eingaben



## Sources

(*ECSDeployAction*/Inputs/Sources)

(Erforderlich, wenn Ihre Aufgabendefinitionsdatei in einem Quell-Repository gespeichert ist)

Wenn Ihre Aufgabendefinitionsdatei in einem Quell-Repository gespeichert ist, geben Sie die Bezeichnung dieses Quell-Repositorys an. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`.

Wenn Ihre Aufgabendefinitionsdatei nicht in einem Quell-Repository enthalten ist, muss sie sich in einem Artefakt befinden, das von einer anderen Aktion generiert wird.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

(*ECSDeployAction*/Inputs/Artifacts)

(Erforderlich, wenn Ihre Aufgabendefinitionsdatei in einem [Ausgabeartefakt](#) aus einer vorherigen Aktion gespeichert ist)

Wenn die Aufgabendefinitionsdatei, die Sie bereitstellen möchten, in einem Artefakt enthalten ist, das von einer vorherigen Aktion generiert wurde, geben Sie dieses Artefakt hier an. Wenn Ihre Aufgabendefinitionsdatei nicht in einem Artefakt enthalten ist, muss sie sich in Ihrem Quell-Repository befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte der Konfiguration – optional

## Configuration

(*ECSDeployAction*/Configuration)

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## region

(Configuration/region)

(Erforderlich)

Geben Sie die AWS Region an, in der sich Ihr Amazon-ECS-Cluster und -Service befinden. Eine Liste der Regionscodes finden Sie unter [Regionale Endpunkte](#) im Allgemeine AWS-Referenz.

Entsprechende Benutzeroberfläche: Registerkarte/Region Konfiguration

## cluster

(*ECSDeployAction*/Configuration/cluster)

(Erforderlich)

Geben Sie den Namen eines vorhandenen Amazon-ECS-Clusters an. Die Aktion Deploy to Amazon ECS stellt Ihre containerisierte Anwendung als Aufgabe in diesem Cluster bereit. Weitere Informationen zu Amazon-ECS-Clustern finden Sie unter [Cluster](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Cluster „Konfiguration“

## service

(*ECSDeployAction*/Configuration/service)

(Erforderlich)

Geben Sie den Namen eines vorhandenen Amazon-ECS-Service an, der die Aufgabendefinitionsdatei instanziiert. Dieser Service muss sich unter dem im `cluster` Feld angegebenen Cluster befinden. Weitere Informationen zu Amazon-ECS-Services finden Sie unter [Amazon-ECS-Services](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Service „Konfiguration“

## task-definition

(*ECSDeployAction*/Configuration/task-definition)

(Erforderlich)

Geben Sie den Pfad zu einer vorhandenen Aufgabendefinitionsdatei an. Wenn sich die Datei in Ihrem Quell-Repository befindet, ist der Pfad relativ zum Stammordner des Quell-Repositorys. Wenn sich Ihre Datei in einem Artefakt aus einer vorherigen Workflow-Aktion befindet, bezieht sich der Pfad auf den Stammordner des Artefakts. Weitere Informationen zu Aufgabendefinitionsdateien finden Sie unter [Aufgabendefinitionen](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Aufgabendefinition Konfiguration

force-new-deployment

(*ECSDeployAction*/Configuration/force-new-deployment)

(Erforderlich)

Wenn diese Option aktiviert ist, kann der Amazon-ECS-Service neue Bereitstellungen ohne Änderungen der Servicedefinition starten. Durch das Erzwingen einer Bereitstellung stoppt der Service alle derzeit ausgeführten Aufgaben und startet neue Aufgaben. Weitere Informationen zum Erzwingen neuer Bereitstellungen finden Sie unter [Aktualisieren eines Services](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Standard: false


Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Neue Bereitstellung des Services erzwingen

codedeploy-appspec

(*ECSDeployAction*/Configuration/codedeploy-appspec)

(Erforderlich, wenn Sie Ihren Amazon-ECS-Service für die Verwendung von Blau/Grün-Bereitstellungen konfiguriert haben, andernfalls lassen Sie ihn weg)

Geben Sie den Namen und den Pfad zu einer vorhandenen CodeDeploy Anwendungsspezifikationsdatei (AppSpec) an. Diese Datei muss sich im Stammverzeichnis Ihres CodeCatalyst Quell-Repositorys befinden. Weitere Informationen zu - AppSpec Dateien finden Sie unter [CodeDeploy Anwendungsspezifikationsdateien \(AppSpec\)](#) im AWS CodeDeploy - Benutzerhandbuch.

 Note

Geben Sie nur CodeDeploy dann Informationen an, wenn Sie Ihren Amazon-ECS-Service so konfiguriert haben, dass er Blau/Grün-Bereitstellungen durchführt. Lassen Sie bei

Bereitstellungen für fortlaufende Updates (Standard) CodeDeploy Informationen weg. Weitere Informationen zu Amazon-ECS-Bereitstellungen finden Sie unter [Amazon-ECS-Bereitstellungstypen](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

 Note

Die CodeDeploy Felder können im visuellen Editor ausgeblendet werden. Informationen dazu, wie sie angezeigt werden, finden Sie unter [Warum fehlen CodeDeploy Felder im Visual Editor?](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/CodeDeploy AppSpec

`codedeploy-application`

(*ECSDeployAction*/Konfiguration/codedeploy-application)

(Erforderlich, wenn enthalten `codedeploy-appspec` ist)

Geben Sie den Namen einer vorhandenen CodeDeploy Anwendung an. Weitere Informationen zu CodeDeploy Anwendungen finden Sie unter [Arbeiten mit Anwendungen in CodeDeploy](#) im AWS CodeDeploy -Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/CodeDeploy Anwendung „Konfiguration“

`codedeploy-deployment-group`

(*ECSDeployAction*/Konfiguration/codedeploy-deployment-group)

(Erforderlich, wenn enthalten `codedeploy-appspec` ist)

Geben Sie den Namen einer vorhandenen CodeDeploy Bereitstellungsgruppe an. Weitere Informationen zu CodeDeploy Bereitstellungsgruppen finden Sie unter [Arbeiten mit Bereitstellungsgruppen in CodeDeploy](#) im AWS CodeDeploy -Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/CodeDeploy Bereitstellungsgruppe „Konfiguration“

`codedeploy-deployment-description`

(*ECSDeployAction*/Konfiguration/codedeploy-deployment-description)

(Optional)

Geben Sie eine Beschreibung der Bereitstellung an, die diese Aktion erstellt. Weitere Informationen finden Sie unter [Arbeiten mit Bereitstellungen in CodeDeploy](#) im AWS CodeDeploy - Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Konfigurations-Registerkarte/CodeDeploy  
Bereitstellungsbeschreibung

## Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“

Im Folgenden finden Sie die YAML-Referenz für die Aktionsdefinition für die Cluster-Aktion Deploy to Kubernetes. Informationen zur Verwendung dieser Aktion finden Sie unter [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das -Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau 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

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: DeployToKubernetesCluster\_nn.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Anzeigename der Aktion

### Identifizier

(*DeployToKubernetesCluster*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: aws/kubernetes-deploy@v1.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/DeployToKubernetesCluster\_nn/aws/kubernetes-deploy@v1-Bezeichnung

### DependsOn

(*DeployToKubernetesCluster*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

*(DeployToKubernetesCluster/Compute)*

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

*(DeployToKubernetesCluster/Compute/Type)*

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsausführungen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungstyp

## Fleet

(*DeployToKubernetesCluster*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen Fleet wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Erweitert – optional/Datenverarbeitungsflotte

## Timeout

(*DeployToKubernetesCluster*/Timeout)

(Optional)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Environment

(*DeployToKubernetesCluster*/Environment)



(Erforderlich)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

Name

*(DeployToKubernetesCluster/Environment/Name)*

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

Connections

*(DeployToKubernetesCluster/Environment/Connections)*

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können unter maximal eine Kontoverbindung angeben `Environment`.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

Name

*(DeployToKubernetesCluster/Environment/Connections/Name)*

(Erforderlich)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Role

(*DeployToKubernetesCluster*/Environment/Connections/Role)

(Erforderlich)

Geben Sie den Namen der IAM-Rolle an, die die Cluster-Aktion Deploy to Kubernetes für den Zugriff auf verwendetAWS. Stellen Sie sicher, dass diese Rolle die folgenden Richtlinien enthält:

- Die folgende Berechtigungsrichtlinie:

### Warning

Beschränken Sie die Berechtigungen auf die in der folgenden Richtlinie aufgeführten. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:ListClusters"
      ],
      "Resource": "*"
    }
  ]
}
```

### Note

Wenn die Rolle zum ersten Mal verwendet wird, verwenden Sie den folgenden Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach ihrer Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- Die folgende benutzerdefinierte Vertrauensrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Stellen Sie sicher, dass diese Rolle hinzugefügt wird:

- Ihre Kontoverbindung. Weitere Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).
- Ihre Kubernetes- ConfigMap. Weitere Informationen zum Hinzufügen einer IAM-ConfigMapRolle zu einem finden Sie unter [Verwalten von IAM-Benutzern und -Rollen](#) in der `-eksctl`Dokumentation.

 Tip

Anweisungen [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#) zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung und finden Sie unter ConfigMap.

**Note**

Sie können den Namen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle hier angeben, wenn Sie möchten. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt. Wenn Sie sich für die Verwendung der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle entscheiden, fügen Sie sie unbedingt zu Ihrer ConfigMap Datei hinzu, indem Sie den Anweisungen unter [IAM-Benutzer und -Rollen verwalten](#) in der `-eksctl` Dokumentation folgen.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Rolle

**Inputs**

*(DeployToKubernetesCluster/Inputs)*

(Optional)

Der Inputs Abschnitt definiert die Daten, die der während einer Workflow-Ausführung `DeployToKubernetesCluster` benötigt.

**Note**

Pro Aktion `Deploy to Amazon EKS` ist nur eine Eingabe (entweder eine Quelle oder ein Artefakt) zulässig.

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

**Sources**

*(DeployToKubernetesCluster/Inputs/Sources)*

(Erforderlich, wenn Ihre Manifestdatei in einem Quell-Repository gespeichert ist)

Wenn Ihre Kubernetes-Manifestdatei oder -Dateien in einem Quell-Repository gespeichert sind, geben Sie die Bezeichnung dieses Quell-Repositorys an. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`.

Wenn Ihre Manifestdateien nicht in einem Quell-Repository enthalten sind, müssen sie sich in einem Artefakt befinden, das von einer anderen Aktion generiert wird.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

*(`DeployToKubernetesCluster`/Inputs/Artifacts)*

(Erforderlich, wenn Ihre Manifestdatei in einem [Ausgabeartefakt](#) aus einer vorherigen Aktion gespeichert ist)

Wenn die Kubernetes-Manifestdatei oder die Kubernetes-Manifestdateien in einem Artefakt enthalten sind, das von einer vorherigen Aktion generiert wurde, geben Sie dieses Artefakt hier an. Wenn Ihre Manifestdateien nicht in einem Artefakt enthalten sind, müssen sie sich in Ihrem Quell-Repository befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte der Konfiguration – optional

## Configuration

*(`DeployToKubernetesCluster`/Configuration)*

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## Namespace

*(`DeployToKubernetesCluster`/Configuration/namespace)*

(Optional)

Geben Sie den Kubernetes-Namespace an, in dem Ihre Kubernetes-Anwendung bereitgestellt wird. Verwenden Sie `default`, wenn Sie keine Namespaces mit Ihrem Cluster verwenden. Weitere Informationen zu Namespaces finden Sie unter [Unterteilen Ihres Clusters mithilfe von Kubernetes-Namespaces](#) in der Kubernetes-Dokumentation.

Wenn Sie den Namespace weglassen, `default` wird der Wert verwendet.

Entsprechende Benutzeroberfläche: Registerkarte/namespace „Konfiguration“

## Region

*(DeployToKubernetesCluster/Configuration/Region)*

(Erforderlich)

Geben Sie die AWS Region an, in der sich Ihr Amazon-EKS-Cluster und -Service befinden. Eine Liste der Regionscodes finden Sie unter [Regionale Endpunkte](#) im Allgemeine AWS-Referenz.

Entsprechende Benutzeroberfläche: Registerkarte/Region Konfiguration

## Cluster

*(DeployToKubernetesCluster/Configuration/Cluster)*

(Erforderlich)

Geben Sie den Namen eines vorhandenen Amazon-EKS-Clusters an. Die Aktion In Kubernetes-Cluster bereitstellen stellt Ihre containerisierte Anwendung in diesem Cluster bereit. Weitere Informationen zu Amazon-EKS-Clustern finden Sie unter [Cluster](#) im Amazon-EKS-Benutzerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Cluster „Konfiguration“

## Manifests

*(DeployToKubernetesCluster/Configuration/Manifests)*

(Erforderlich)

Geben Sie in der Kubernetes-Dokumentation den Pfad zu Ihrer/Ihren YAML-formatierten Kubernetes-Manifestdatei(en) an, die als Konfigurationsdateien, Konfigurationsdateien oder einfache Konfigurationen bezeichnet werden.

Wenn Sie mehrere Manifestdateien verwenden, platzieren Sie sie in einem einzelnen Ordner und verweisen Sie auf diesen Ordner. Manifestdateien werden von Kubernetes alphanumerisch verarbeitet. Stellen Sie daher sicher, dass Dateinamen zunehmende Zahlen oder Buchstaben vorangestellt werden, um die Verarbeitungsreihenfolge zu steuern. Beispielsweise:

```
00-namespace.yaml
```

```
01-deployment.yaml
```

Wenn sich Ihre Manifestdateien in Ihrem Quell-Repository befinden, ist der Pfad relativ zum Stammordner des Quell-Repositorys. Wenn sich die Dateien in einem Artefakt aus einer vorherigen Workflow-Aktion befinden, bezieht sich der Pfad auf den Stammordner des Artefakts.


Beispiele:

```
Manifests/
```

```
deployment.yaml
```

```
my-deployment.yml
```

Verwenden Sie keine Platzhalter (\*).

 Note

[Helm-Charts](#) und [Kustomisierungsdateien](#) werden nicht unterstützt.

Weitere Informationen zu Manifestdateien finden Sie unter [Organisieren von Ressourcenkonfigurationen](#) in der Kubernetes-Dokumentation.

Entsprechende Benutzeroberfläche: Registerkarte/Manifests für Konfiguration

## Referenz zur Aktion „GitHub Aktionen“

Im Folgenden finden Sie die YAML-Referenz der Aktionsdefinition für die Aktion GitHub Aktionen.

Wählen Sie im folgenden Code eine YAML-Eigenschaft aus, um eine Beschreibung anzuzeigen.

 Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das -Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau 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*

**Configuration****Steps:**

- *github-actions-code*

## Aktionsname

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Entsprechende Benutzeroberfläche: Registerkarte/*Aktionsname* der Konfiguration

## Identifizier

(*Aktionsname* /Identifizier)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Verwenden Sie `aws/github-actions-runner@v1` für GitHub Aktionen.

Entsprechende Benutzeroberfläche: Workflow-*Diagramm/Aktionsname* /aws/github-actions-runner@v1-Bezeichnung

## DependsOn

(*Aktionsname* /DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

(*Aktionsname* /Compute)

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Fleet

*(Aktionsname)*/Compute/Fleet

(Optional)

Geben Sie die Maschine oder Flotte an, auf der Ihr Workflow oder Ihre Workflow-Aktionen ausgeführt werden. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen Fleet wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Datenverarbeitungsflotte – optional

## Timeout

*(Aktionsname)* /Timeout

(Optional)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt

5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Environment

(*Aktionsname* /Environment)

(Optional)

Geben Sie die CodeCatalyst Umgebung an, die mit der Aktion verwendet werden soll.

Weitere Informationen zu Umgebungen finden Sie unter [Arbeiten mit Umgebungen](#) und [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: Registerkarte/Umgebung/Konto/Rolle der Konfiguration

## Name

(*Aktionsname* /Environment/Name)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie den Namen einer vorhandenen Umgebung an, die Sie der Aktion zuordnen möchten.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Umgebung

## Connections

(*Aktionsname* /Environment/Connections)

(Erforderlich, wenn enthalten [Environment](#) ist)

Geben Sie die Kontoverbindung an, die der Aktion zugeordnet werden soll. Sie können unter maximal eine Kontoverbindung angeben `Environment`.

Weitere Informationen zu Kontoverbindungen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#). Informationen zum Zuordnen einer Kontoverbindung zu Ihrer Umgebung finden Sie unter [Erstellen einer Umgebung](#).

Entsprechende Benutzeroberfläche: keine

## Name

(*Aktionsname*)/Environment/Connections/Name

(Optional)

Geben Sie den Namen der Kontoverbindung an.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/AWSKontoverbindung

## Role

(*Aktionsname*)/Environment/Connections/Role

(Optional)

Geben Sie den Namen der IAM-Rolle an, die diese Aktion verwendet, um auf -AWSServices wie Amazon S3 und Amazon ECR zuzugreifen und diese auszuführen. Stellen Sie sicher, dass diese Rolle zu Ihrer Kontoverbindung hinzugefügt wird. Informationen zum Hinzufügen einer IAM-Rolle zu einer Kontoverbindung finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

### Note

Möglicherweise können Sie hier den Namen der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle angeben, sofern sie über ausreichende Berechtigungen verfügt. Weitere Informationen über diese Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Beachten Sie, dass die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle über sehr umfassende Berechtigungen verfügt, die ein Sicherheitsrisiko darstellen können. Wir empfehlen, diese Rolle nur in Tutorials und Szenarien zu verwenden, in denen die Sicherheit weniger ein Problem darstellt.

### Warning

Beschränken Sie die Berechtigungen auf die Berechtigungen, die für die GitHub Aktion erforderlich sind. Die Verwendung einer Rolle mit umfassenderen Berechtigungen kann ein Sicherheitsrisiko darstellen.

Entsprechende Benutzeroberfläche: Registerkarte „Konfiguration“/„Umgebung/Konto/Rolle“/Rolle

## Inputs

(*Aktionsname* /Inputs)

(Optional)

Der Inputs Abschnitt definiert die Daten, die eine Aktion während einer Workflow-Ausführung benötigt.

### Note

Pro GitHub Aktionsaktion sind maximal vier Eingaben (eine Quelle und drei Artefakte) zulässig. Variablen werden nicht auf diese Summe angerechnet.

Wenn Sie auf Dateien verweisen müssen, die sich in verschiedenen Eingaben befinden (z. B. eine Quelle und ein Artefakt), ist die Quelleingabe die primäre Eingabe und das Artefakt ist die sekundäre Eingabe. Verweise auf Dateien in sekundären Eingaben verwenden ein spezielles Präfix, um sie vom primären zu trennen. Details hierzu finden Sie unter [Beispiel: Referenzieren von Dateien in mehreren Artefakten](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

## Sources

(*Aktionsname*)/Inputs/Sources

(Optional)

Geben Sie die Bezeichnungen an, die die Quell-Repositorys darstellen, die von der Aktion benötigt werden. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`, die das Quell-Repository darstellt, in dem Ihre Workflow-Definitionsdatei gespeichert ist.

Wenn Sie eine Quelle weglassen, müssen Sie mindestens ein Eingabeartefakt unter angeben *action-name*/Inputs/Artifacts.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

*(Aktionsname)*/Inputs/Artifacts

(Optional)

Geben Sie Artefakte aus früheren Aktionen an, die Sie als Eingabe für diese Aktion bereitstellen möchten. Diese Artefakte müssen in früheren Aktionen bereits als Ausgabeartefakte definiert sein.

Wenn Sie keine Eingabeartefakte angeben, müssen Sie mindestens ein Quell-Repository unter angeben *action-name*/Inputs/Sources.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

### Note

Wenn die Dropdown-Liste Artefakte – optional nicht verfügbar ist (visueller Editor) oder wenn Sie bei der Validierung Ihrer YAML (YAML-Editor) Fehler in erhalten, kann dies daran liegen, dass die Aktion nur eine Eingabe unterstützt. Versuchen Sie in diesem Fall, die Quelleingabe zu entfernen.

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte – optional

## Variables - input

*(Aktionsname)* /Inputs/Variables)

(Optional)

Geben Sie eine Folge von Name/Wert-Paaren an, die die Eingabevariablen definieren, die Sie der Aktion zur Verfügung stellen möchten. Variablennamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Variablennamen zu aktivieren.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Variablen Eingaben – optional

## Outputs

*(Aktionsname /Outputs)*

(Optional)

Definiert die Daten, die von der Aktion während einer Workflow-Ausführung ausgegeben werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“

## Artifacts - output

*(Aktionsname)/Outputs/Artifacts*

(Optional)

Geben Sie den Namen eines Artefakts an, das von der Aktion generiert wurde. Artefaktnamen müssen innerhalb eines Workflows eindeutig und auf alphanumerische Zeichen (a-z, A-Z, 0-9) und Unterstriche (\_) beschränkt sein. Leerzeichen, Bindestriche (-) und andere Sonderzeichen sind nicht zulässig. Sie können keine Anführungszeichen verwenden, um Leerzeichen, Bindestriche und andere Sonderzeichen in Ausgabeartefaktnamen zu aktivieren.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artefakte für Ausgaben

## Name

*(Aktionsname /Outputs/Artifacts/Name)*

(Erforderlich, wenn enthalten [Artifacts - output](#) ist)

Geben Sie den Namen eines Artefakts an, das von der Aktion generiert wurde. Artefaktnamen müssen innerhalb eines Workflows eindeutig und auf alphanumerische Zeichen (a-z, A-Z, 0-9) und Unterstriche (\_) beschränkt sein. Leerzeichen, Bindestriche (-) und andere Sonderzeichen sind nicht zulässig. Sie können keine Anführungszeichen verwenden, um Leerzeichen, Bindestriche und andere Sonderzeichen in Ausgabeartefaktnamen zu aktivieren.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).



Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Artifacts/Artefakt hinzufügen/Artefaktnamen erstellen

## Files

(*Aktionsname* /Outputs/Artifacts/Files)

(Erforderlich, wenn enthalten [Artifacts - output](#) ist)

Geben Sie die Dateien an, die in das Artefakt CodeCatalyst eingeschließt, das von der Aktion ausgegeben wird. Diese Dateien werden von der Workflow-Aktion generiert, wenn sie ausgeführt wird, und sind auch in Ihrem Quell-Repository verfügbar. Dateipfade können sich in einem Quell-Repository oder einem Artefakt aus einer vorherigen Aktion befinden und relativ zum Quell-Repository oder Artefaktstamm sein. Sie können globale Muster verwenden, um Pfade anzugeben. Beispiele:

- Um eine einzelne Datei anzugeben, die sich im Stammverzeichnis Ihres Build-Speicherorts oder Quell-Repository-Speicherorts befindet, verwenden Sie `my-file.jar`.
- Um eine einzelne Datei in einem Unterverzeichnis anzugeben, verwenden Sie `directory/my-file.jar` oder `directory/subdirectory/my-file.jar`.
- Um alle Dateien anzugeben, verwenden Sie `**/*`. Das `**` Glob-Muster zeigt an, dass mit einer beliebigen Anzahl von Unterverzeichnissen übereinstimmt.
- Um alle Dateien und Verzeichnisse in einem Verzeichnis mit dem Namen `directory`, verwenden Sie `directory/**/*`. Das `**` Glob-Muster zeigt an, dass mit einer beliebigen Anzahl von Unterverzeichnissen übereinstimmt.
- Um alle Dateien in einem Verzeichnis mit dem Namen `directory`, aber keines seiner Unterverzeichnisse, verwenden Sie `directory/*`.

### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder ein anderes Sonderzeichen enthält, schließen Sie den Pfad mit doppelten Anführungszeichen (") ein. Weitere Informationen zu Sonderzeichen finden Sie unter [Richtlinien und Konventionen zur Syntax](#).

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

**Note**

Möglicherweise müssen Sie dem Dateipfad ein Präfix hinzufügen, um anzugeben, in welchem Artefakt oder welcher Quelle es gefunden werden soll. Weitere Informationen finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Artifacts/Hinzufügen von Artefakten/Dateien, die vom Build erstellt wurden

**Variables - output**

*(Aktionsname)*/Outputs/Variables

(Optional)

Geben Sie die Variablen an, die die Aktion exportieren soll, damit sie für nachfolgende Aktionen verfügbar sind.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Variablen/Variable hinzufügen

variable-name-1

*(action-name)*/Outputs/Variables variable-name-1)

(Optional)

Geben Sie den Namen einer Variablen an, die die Aktion exportieren soll. Diese Variable muss bereits im Steps Abschnitt Inputs oder derselben Aktion definiert sein.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Variablen/Variable/Variable/Name hinzufügen

**AutoDiscoverReports**

*(Aktionsname)*/Outputs/AutoDiscoverReports

(Optional)

Definiert die Konfiguration für das Feature zur automatischen Erkennung.

Wenn Sie die automatische Erkennung aktivieren, CodeCatalyst durchsucht alle an die Aktion `Inputs` übergebenen Dateien sowie alle Dateien, die von der Aktion selbst generiert wurden, und sucht nach Test-, Codeabdeckungs- und Software-Kompositionsanalyse (SCA)-Berichten. Für jeden gefundenen Bericht CodeCatalyst transformiert ihn in einen CodeCatalyst Bericht. Ein CodeCatalyst Bericht ist ein Bericht, der vollständig in den CodeCatalyst Service integriert ist und über die CodeCatalyst Konsole angezeigt und bearbeitet werden kann.

 Note

Standardmäßig überprüft die automatische Erkennungsfunktion alle Dateien. Sie können einschränken, welche Dateien mit den [ExcludePaths](#) Eigenschaften [IncludePaths](#) oder überprüft werden.

Entsprechende Benutzeroberfläche: keine

Enabled

*(Aktionsname)*/Outputs/AutoDiscoverReports/Enabled

(Optional)

Aktivieren oder deaktivieren Sie die Funktion zur automatischen Erkennung.

Gültige Werte sind `true` oder `false`.

Wenn weggelassen `Enabled` wird, ist der Standardwert `true`.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“/ „Berichte“/ „Automatisch erkennen“

ReportNamePrefix

*(Aktionsname)*/Outputs/AutoDiscoverReports/ReportNamePrefix

(Erforderlich, wenn enthalten und aktiviert [AutoDiscoverReports](#) ist)

Geben Sie ein Präfix an, das allen gefundenen Berichten CodeCatalyst vorangestellt wird, um die zugehörigen CodeCatalyst Berichte zu benennen. Wenn Sie beispielsweise das Präfix

angeben `AutoDiscovered` und CodeCatalyst zwei Testberichte, `TestSuiteOne.xml` und , automatisch erkennt `TestSuiteTwo.xml`, werden die zugehörigen CodeCatalyst Berichte als `AutoDiscoveredTestSuiteOne` und bezeichnet `AutoDiscoveredTestSuiteTwo`.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“/„Berichte“/„Berichte automatisch erkennen“/„Berichtspräfix“

## IncludePaths

*(Aktionsname)*/Outputs/AutoDiscoverReports/IncludePaths

Oder

*(action-name)*/Outputs/Reports/ *report-name-1*/IncludePaths)

(Erforderlich, wenn enthalten und aktiviert [AutoDiscoverReports](#) ist, oder , wenn enthalten [Reports](#) ist)

Geben Sie die Dateien und Dateipfade an, die bei der Suche nach Rohberichten CodeCatalyst enthält. Wenn Sie beispielsweise angeben, CodeCatalyst durchsucht das gesamte `Build-Image"/test/report/*"`, das von der Aktion verwendet wird, und sucht nach dem `/test/report/*` Verzeichnis . Wenn dieses Verzeichnis gefunden wird, sucht CodeCatalyst dann nach Berichten in diesem Verzeichnis.

### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder andere Sonderzeichen enthält, schließen Sie den Pfad in doppelte Anführungszeichen (") ein"". Weitere Informationen zu Sonderzeichen finden Sie unter [Richtlinien und Konventionen zur Syntax](#).

Wenn diese Eigenschaft weggelassen wird, ist der Standardwert , was bedeutet `**/*"`, dass die Suche alle Dateien auf allen Pfaden enthält.

### Note

Bei manuell konfigurierten Berichten `IncludePaths` muss ein Glob-Muster sein, das einer einzelnen Datei entspricht.

Entsprechende Benutzeroberfläche:

- Ausgabe-Registerkarte/Berichte/Automatisch Berichte erkennen/Pfade einschließen/ausschließen/Pfade einschließen
- Ausgabe-Registerkarte/Berichte/ Manuelles Konfigurieren von Berichten/*Berichtsname-1*'Einschließen/Ausschließen von Pfaden'/Einschließen von Pfaden

## ExcludePaths

*(Aktionsname)*/Outputs/AutoDiscoverReports/ExcludePaths

Oder

*(action-name)*/Outputs/Reports/ *report-name-1*/ExcludePaths)

(Optional)

Geben Sie die Dateien und Dateipfade an, die bei der Suche nach Rohberichten CodeCatalyst ausschließt. Wenn Sie beispielsweise angeben `/test/my-reports/**/*`, CodeCatalyst sucht nicht nach Dateien im `/test/my-reports/` Verzeichnis . Um alle Dateien in einem Verzeichnis zu ignorieren, verwenden Sie das `**/*` Glob-Muster.

### Note

Wenn Ihr Dateipfad ein oder mehrere Sternchen (\*) oder andere Sonderzeichen enthält, schließen Sie den Pfad in doppelte Anführungszeichen (") ein"". Weitere Informationen zu Sonderzeichen finden Sie unter [Richtlinien und Konventionen zur Syntax](#).

Entsprechende Benutzeroberfläche:

- Ausgabe-Registerkarte/Berichte/Automatisch Berichte erkennen/Pfade einschließen/ausschließen/Pfade ausschließen
- Ausgabe-Registerkarte/Berichte/ Manuelles Konfigurieren von Berichten/*Berichtsname-1*/„Einschließen/Ausschließen von Pfaden“/Ausschließen von Pfaden

## SuccessCriteria

*(Aktionsname)*/Outputs/AutoDiscoverReports/SuccessCriteria

Oder

(*action-name*/Outputs/Reports/ *report-name-1*/SuccessCriteria)

(Optional)

Geben Sie die Erfolgskriterien für die Berichte Test, Codeabdeckung, Software-Kompositionsanalyse (SCA) und statische Analyse (SA) an.

Weitere Informationen finden Sie unter [Erfolgskriterien für Berichte konfigurieren](#).

Entsprechende Benutzeroberfläche:

- Registerkarte „Ausgaben“/ „Berichte/Erfolgskriterien automatisch erkennen“
- Registerkarte „Outputs“/Berichte/ Manuelles Konfigurieren von Berichten/*Berichtsname-1*/Erfolgskriterien

## PassRate

(*Aktionsname* /Outputs/AutoDiscoverReports/SuccessCriteria/PassRate)

Oder

(*action-name*/Outputs/Reports/ *report-name-1*/SuccessCriteria/PassRate)

(Optional)

Geben Sie den Prozentsatz der Tests in einem Testbericht an, die bestehen müssen, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Gültige Werte sind Dezimalzahlen. Zum Beispiel: 50, 60 . 5. Die Kriterien für die Passrate werden nur auf Testberichte angewendet. Weitere Informationen zu Testberichten finden Sie unter [Testberichte](#).

Entsprechende Benutzeroberfläche:

- Registerkarte „Outputs“/ „Reports“/ „Automatisch Berichte erkennen“/ „Erfolgskriterien“/ „Passrate“
- Registerkarte „Outputs“/Berichte/ Manuelles Konfigurieren von Berichten/*Berichtsname-1*/ Erfolgskriterien/Passrate

## LineCoverage

(*Aktionsname*)/Outputs/AutoDiscoverReports/SuccessCriteria/LineCoverage

Oder

(*action-name*/Outputs/Reports/ *report-name-1*/SuccessCriteria/LineCoverage)

(Optional)

Geben Sie den Prozentsatz der Zeilen in einem Codeabdeckungsbericht an, der abgedeckt werden muss, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Gültige Werte sind Dezimalzahlen. Zum Beispiel: 50, 60 . 5. Kriterien für die Zeilenabdeckung werden nur auf Codeabdeckungsberichte angewendet. Weitere Informationen zu Codeabdeckungsberichten finden Sie unter [Berichte zur Codeabdeckung](#).

Entsprechende Benutzeroberfläche:

- Registerkarte Ausgaben/Berichte/Automatisch Berichte erkennen/Erfolgskriterien/Zeilenabdeckung
- Registerkarte „Outputs“/ „Reports“/ „Manuelle Konfiguration von Berichten/*Berichtsname-1*/Erfolgskriterien/Zeilenabdeckung“

## BranchCoverage

(*Aktionsname* /Outputs/AutoDiscoverReports/SuccessCriteria/BranchCoverage)

Oder

(*action-name*/Outputs/Reports/ *report-name-1*/SuccessCriteria/BranchCoverage)

(Optional)

Geben Sie den Prozentsatz der Verzweigungen in einem Code-Abdeckungsbericht an, der abgedeckt werden muss, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Gültige Werte sind Dezimalzahlen. Zum Beispiel: 50, 60 . 5. Kriterien für die Verzweigung werden nur auf Code-Abdeckungsberichte angewendet. Weitere Informationen zu Codeabdeckungsberichten finden Sie unter [Berichte zur Codeabdeckung](#).

Entsprechende Benutzeroberfläche:

- Registerkarte „Outputs“/ „Reports“/ „Automatisch Berichte erkennen“/ „Erfolgskriterien“/ „Branch-Abdeckung“
- Registerkarte „Outputs“/Berichte/ Manuelles Konfigurieren von Berichten/*Berichtsname-1*/Erfolgskriterien/Branch-Abdeckung

## Vulnerabilities

*(Aktionsname)*/Outputs/AutoDiscoverReports/SuccessCriteria/Vulnerabilities

Oder

*(action-name)*/Outputs/Reports/ *report-name-1*/SuccessCriteria/Vulnerabilities)

(Optional)

Geben Sie die maximale Anzahl und den Schweregrad von Schwachstellen an, die im SCA-Bericht zulässig sind, damit der zugehörige CodeCatalyst Bericht als bestanden markiert wird. Um Schwachstellen anzugeben, müssen Sie Folgendes angeben:

- Der Mindestschweregrad der Schwachstellen, die Sie in die Anzahl aufnehmen möchten. Gültige Werte, von den meisten bis zu den geringsten Schweregraden, sind: CRITICAL, HIGH, MEDIUM, LOW, INFORMATIONAL.

Wenn Sie beispielsweise wählen HIGH, werden - HIGH und -CRITICAL Schwachstellen gemildert.

- Die maximale Anzahl von Schwachstellen des angegebenen Schweregrads, die Sie zulassen möchten. Wenn Sie diese Zahl überschreiten, wird der CodeCatalyst Bericht als fehlgeschlagen markiert. Gültige Werte sind ganze Zahlen.

Schwachstellenkriterien werden nur auf SCA-Berichte angewendet. Weitere Informationen zu SCA-Berichten finden Sie unter [Berichte zur Analyse der Softwarezusammensetzung](#).

Um den minimalen Schweregrad anzugeben, verwenden Sie die `-Severity`Eigenschaft. Um die maximale Anzahl von Schwachstellen anzugeben, verwenden Sie die `-Number`Eigenschaft.

Weitere Informationen zu SCA-Berichten finden Sie unter [Typen von Testberichten](#).

Entsprechende Benutzeroberfläche:

- Registerkarte „Outputs“/ „Reports“/ „Automatisch Berichte erkennen“/ „Erfolgskriterien/Schwachstellen“
- Registerkarte „Outputs“/ „Reports“/ „Manuelles Konfigurieren von Berichten/*Berichtsname-1*/Erfolgskriterien/Schwachstellen“



## Reports

(*Aktionsname*/Outputs/Reports )

(Optional)

Ein Abschnitt, der die Konfiguration für Testberichte angibt.

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Berichte

report-name-1

(*action-name*/Outputs/Reports/ report-name-1)

(Erforderlich, wenn enthalten [Reports](#) ist)

Der Name, den Sie dem CodeCatalyst Bericht geben möchten, der aus Ihren Rohberichten generiert wird.

Entsprechende Benutzeroberfläche: Registerkarte/Berichte/ Manuelles Konfigurieren von Berichten/Berichtsname

## Format

(*action-name*/Outputs/Reports/ *report-name-1*/Format)

(Erforderlich, wenn enthalten [Reports](#) ist)

Geben Sie das Dateiformat an, das Sie für Ihre Berichte verwenden. Mögliche Werte können sein wie folgt.

- Für Testberichte:
  - Geben Sie für Cucumber JSON Cucumber (visueller Editor) oder CUCUMBERJSON (YAML-Editor) an.
  - Geben Sie für JUnit XML JUnit (visueller Editor) oder JUNITXML (YAML-Editor) an.
  - Geben Sie für NUnit XML NUnit (visueller Editor) oder NUNITXML (YAML-Editor) an.
  - Geben Sie für NUnit 3 XML NUnit3 (visueller Editor) oder NUNIT3XML (YAML-Editor) an.
  - Geben Sie für Visual Studio TRX Visual Studio TRX (visueller Editor) oder VISUALSTUDIOTRX (YAML-Editor) an.

- Geben Sie für TestNG XML TestNG (visueller Editor) oder TESTNGXML (YAML-Editor) an.
- Für Code-Abdeckungsberichte:
  - Geben Sie für Clover XML (visueller Editor) oder CLOVERXML (YAML-Editor) an.
  - Geben Sie für CoSpeedura XML CoSpeedura (visueller Editor) oder COBERTURAXML (YAML-Editor) an.
  - Geben Sie für JaCoCo XML JaCoCo (visueller Editor) oder JACOCOXML (YAML-Editor) an.
  - Geben Sie für von [Simplecov](#) generiertes SimpleCov JSON, nicht [Simplecov-json](#), Simplecov (visueller Editor) oder SIMPLECOV (YAML-Editor) an.
- Für Berichte zur Software-Kompositionsanalyse (SCA):
  - Geben Sie für SARIF SARIF (visueller Editor) oder SARIFSCA (YAML-Editor) an.

Entsprechende Benutzeroberfläche: Registerkarte Ausgaben/Berichte/ Manuelles Konfigurieren von Berichten/Bericht hinzufügen/*Berichtsname-1*/Berichtstyp und Berichtsformat

## Configuration

(*Aktionsname* /Configuration)

(Erforderlich) Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## Steps

(*Aktionsname*)/Configuration/Steps

(Erforderlich)

Geben Sie Ihren GitHub Aktionscode so an, wie er auf der Detailseite der Aktion in [GitHub Marketplace](#) angezeigt wird. Fügen Sie den Code gemäß den folgenden Richtlinien hinzu:

1. Fügen Sie den Code aus dem `steps` : Abschnitt GitHub Aktion in den `Steps` : Abschnitt des CodeCatalyst Workflows ein. Der Code beginnt mit einem Bindestrich (-) und sieht in etwa wie folgt aus.

GitHub Code zum Einfügen:

```
- name: Lint Code Base
```

```
uses: github/super-linter@v4
env:
  VALIDATE_ALL_CODEBASE: false
  DEFAULT_BRANCH: master
  GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

- Überprüfen Sie den Code, den Sie gerade eingefügt haben, und ändern Sie ihn nach Bedarf, damit er den CodeCatalyst Standards entspricht. Mit dem vorherigen Codeblock können Sie beispielsweise den Code in *roter Kursivschrift* entfernen und den Code in fett gedruckter Einfügen.

CodeCatalyst Workflow-yaml:

```
Steps:
  - name: Lint Code Base
    uses: github/super-linter@v4
    env:
      VALIDATE_ALL_CODEBASE: false
      DEFAULT_BRANCH: mastermain
      GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

- Für zusätzlichen Code, der in der GitHub Aktion enthalten ist, aber nicht im `-steps`-Abschnitt vorhanden ist, fügen Sie ihn dem CodeCatalyst Workflow mit dem CodeCatalyst-Klarcodem hinzu. Sie können die überprüfen [Referenz zur Workflow-Definition](#), um einen Einblick in die Portierung Ihres GitHub Codes in zu erhalten CodeCatalyst. Detaillierte Migrationsschritte liegen außerhalb des Rahmens dieses Handbuchs.

Hier ist ein Beispiel für die Angabe von Dateipfaden in einer Aktion GitHub Aktionen:

```
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
```

Weitere Informationen zum Angeben von Dateipfaden finden Sie unter [Verweisen auf Dateien in einem Quell-Repository](#) und [Referenzieren von Dateien in einem Artefakt](#).

Entsprechende Benutzeroberfläche: Konfigurationsregisterkarte/GitHub Aktionen YAML

## Referenz zur Aktion „Amazon-ECS-Aufgabendefinition rendern“

Im Folgenden finden Sie die YAML-Referenz der Aktionsdefinition für die Render-Amazon-ECS-Aufgabendefinitionsaktion. Informationen zur Verwendung dieser Aktion finden Sie unter [Aktion „Amazon ECS-Aufgabendefinition rendern“ hinzufügen](#).

### Note

Die meisten der folgenden YAML-Eigenschaften haben entsprechende UI-Elemente im visuellen Editor. Verwenden Sie Strg+F, um ein UI-Element nachzuschlagen. Das Element wird mit der zugehörigen YAML-Eigenschaft aufgelistet.

```
# The workflow definition starts here.
# See Immobilien auf höchstem Niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSRenderTaskDefinition\_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

(Erforderlich)

Geben Sie den Namen der Aktion an. Alle Aktionsnamen müssen innerhalb des Workflows eindeutig sein. Aktionsnamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Aktionsnamen zu aktivieren.

Standard: `ECSRenderTaskDefinition_nn`.

Entsprechende Benutzeroberfläche: Registerkarte/Aktionsname der Konfiguration

### Identifizier

(*ECSRenderTaskDefinition*/Identifizier)

(Erforderlich)

Identifiziert die Aktion. Ändern Sie diese Eigenschaft nur, wenn Sie die Version ändern möchten. Weitere Informationen finden Sie unter [Mit Action-Versionen arbeiten](#).

Standard: `aws/ecs-render-task-definition@v1`.

Entsprechende Benutzeroberfläche: Workflow-Diagramm/`ECSRenderTaskDefinition_nn/aws/ecs-render-task-definition@v1`-Bezeichnung

### DependsOn

(*ECSRenderTaskDefinition*/DependsOn)

(Optional)

Geben Sie eine Aktion oder Aktionsgruppe an, die erfolgreich ausgeführt werden muss, damit diese Aktion ausgeführt werden kann.

Weitere Informationen zur Funktionalität „abhängig von“ finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

Entsprechende Benutzeroberfläche: Registerkarte Eingaben/Abhängig von – optional

## Compute

*(ECSRenderTaskDefinition/Compute)*

(Optional)

Die Computer-Engine, die zum Ausführen Ihrer Workflow-Aktionen verwendet wird. Sie können die Rechenleistung entweder auf Workflow-Ebene oder auf Aktionsebene angeben, aber nicht auf beides. Wenn auf Workflow-Ebene angegeben, gilt die Datenverarbeitungsconfiguration für alle im Workflow definierten Aktionen. Auf Workflow-Ebene können Sie auch mehrere Aktionen auf derselben Instance ausführen. Weitere Informationen finden Sie unter [Rechenleistung für mehrere Aktionen gemeinsam nutzen](#).

Entsprechende Benutzeroberfläche: keine

## Type

*(ECSRenderTaskDefinition/Compute/Type)*

(Erforderlich, wenn enthalten [Compute](#) ist)

Der Typ der Rechen-Engine. Sie können einen der folgenden Werte verwenden:

- EC2 (visueller Editor) oder EC2 (YAML-Editor)  
Optimiert für Flexibilität bei Aktionsausführungen.
- Lambda (visueller Editor) oder Lambda (YAML-Editor)  
Optimierte Startgeschwindigkeiten für Aktionen.

Weitere Informationen zu Datentypen finden Sie unter [Über Berechnungstypen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Datenverarbeitungstyp „Konfiguration“

## Fleet

(*ECSRenderTaskDefinition*/Compute/Fleet)

(Optional)

Geben Sie die Maschine oder Flotte an, die Ihren Workflow oder Ihre Workflow-Aktionen ausführen wird. Bei On-Demand-Flotten stellt der Workflow beim Start einer Aktion die benötigten Ressourcen bereit und die Maschinen werden nach Abschluss der Aktion zerstört. Beispiele für On-Demand-Flotten: `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Weitere Informationen zu On-Demand-Flotten finden Sie unter [Flotteneigenschaften auf Abruf](#).

Mit bereitgestellten Flotten konfigurieren Sie eine Reihe von dedizierten Maschinen für die Ausführung Ihrer Workflow-Aktionen. Diese Maschinen bleiben inaktiv und können sofort Aktionen verarbeiten. Weitere Informationen zu bereitgestellten Flotten finden Sie unter [Bereitgestellte Flotteneigenschaften](#).

Wenn weggelassen Fleet wird, ist der Standardwert `Linux.x86-64.Large`.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Datenverarbeitungsflotte

## Timeout

(*ECSRenderTaskDefinition*/Timeout)

(Optional)

Geben Sie die Zeit in Minuten (YAML-Editor) oder Stunden und Minuten (visueller Editor) an, die die Aktion ausgeführt werden kann, bevor die Aktion CodeCatalyst beendet. Das Minimum beträgt 5 Minuten und das Maximum ist unter beschrieben [Kontingente für Workflows in CodeCatalyst](#). Das Standard-Timeout entspricht dem maximalen Timeout.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration/Timeout – optional

## Inputs

(*ECSRenderTaskDefinition*/Inputs)

(Optional)

Der `Inputs` Abschnitt definiert die Daten, die während einer Workflow-Ausführung `ECSRenderTaskDefinition` benötigt.

#### Note

Pro `Render-Amazon-ECS-Aufgabendefinitionsaktion` ist nur eine Eingabe (entweder eine Quelle oder ein Artefakt) zulässig. Variablen werden nicht auf diese Summe angerechnet.

Entsprechende Benutzeroberfläche: Registerkarte Eingaben

## Sources

(*`ECSRenderTaskDefinition`*/Inputs/Sources)

(Erforderlich, wenn Ihre Aufgabendefinitionsdatei in einem Quell-Repository gespeichert ist)

Wenn Ihre Aufgabendefinitionsdatei in einem Quell-Repository gespeichert ist, geben Sie die Bezeichnung dieses Quell-Repositorys an. Derzeit ist die einzige unterstützte Bezeichnung `WorkflowSource`.

Wenn Ihre Aufgabendefinitionsdatei nicht in einem Quell-Repository enthalten ist, muss sie sich in einem Artefakt befinden, das von einer anderen Aktion generiert wird.

Weitere Informationen zu Quellen finden Sie unter [Mit Quellen arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte/Quellen – optional

## Artifacts - input

(*`ECSRenderTaskDefinition`*/Inputs/Artifacts)

(Erforderlich, wenn Ihre Aufgabendefinitionsdatei in einem [Ausgabeartefakt](#) aus einer vorherigen Aktion gespeichert ist)

Wenn die Aufgabendefinitionsdatei, die Sie bereitstellen möchten, in einem Artefakt enthalten ist, das von einer vorherigen Aktion generiert wurde, geben Sie dieses Artefakt hier an. Wenn Ihre Aufgabendefinitionsdatei nicht in einem Artefakt enthalten ist, muss sie sich in Ihrem Quell-Repository befinden.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).



Entsprechende Benutzeroberfläche: Registerkarte/Artefakte der Konfiguration – optional

## Variables - input

(*ECSRenderTaskDefinition*/Inputs/Variables)

(Erforderlich)

Geben Sie eine Folge von Name/Wert-Paaren an, die die Eingabevariablen definieren, die Sie der Aktion zur Verfügung stellen möchten. Variablennamen sind auf alphanumerische Zeichen (a-z, A-Z, 0-9), Bindestriche (-) und Unterstriche (\_) beschränkt. Leerzeichen sind nicht erlaubt. Sie können keine Anführungszeichen verwenden, um Sonderzeichen und Leerzeichen in Variablennamen zu aktivieren.

Weitere Informationen zu Variablen, einschließlich Beispielen, finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Variablen Eingaben – optional

## Configuration

(*ECSRenderTaskDefinition*/Configuration)

(Erforderlich)

Ein Abschnitt, in dem Sie die Konfigurationseigenschaften der Aktion definieren können.

Entsprechende Benutzeroberfläche: Registerkarte Konfiguration

## task-definition

(*ECSRenderTaskDefinition*/Configuration/task-definition)

(Erforderlich)

Geben Sie den Pfad zu einer vorhandenen Aufgabendefinitionsdatei an. Wenn sich die Datei in Ihrem Quell-Repository befindet, ist der Pfad relativ zum Stammordner des Quell-Repositorys. Wenn sich Ihre Datei in einem Artefakt aus einer vorherigen Workflow-Aktion befindet, bezieht sich der Pfad auf den Stammordner des Artefakts. Weitere Informationen zu Aufgabendefinitionsdateien finden Sie unter [Aufgabendefinitionen](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Aufgabendefinition Konfiguration

## container-definition-name

(*ECSRenderTaskDefinition*/Configuration/container-definition-name)

(Erforderlich)

Geben Sie den Namen des Containers an, in dem Ihr Docker-Image ausgeführt wird. Sie finden diesen Namen im name Feld `containerDefinitions`, in Ihrer Aufgabendefinitionsdatei. Weitere Informationen finden Sie unter [Name](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Containername der Konfiguration

## image

(*ECSRenderTaskDefinition*/Configuration/image)

(Erforderlich)

Geben Sie den Namen des Docker-Images an, das die Render-Amazon-ECS-Aufgabendefinitionsaktion Ihrer Aufgabendefinitionsdatei hinzufügen soll. Die Aktion fügt diesen Namen dem `image` Feld `containerDefinitions`, in Ihrer Aufgabendefinitionsdatei hinzu. Wenn im `image` Feld bereits ein Wert vorhanden ist, wird dieser von der Aktion überschrieben. Sie können Variablen in den Bildnamen aufnehmen.

Beispiele:

Wenn Sie angeben `MyDockerImage:${WorkflowSource.CommitId}`, fügt die Aktion `MyDockerImage:commit-id` der Aufgabendefinitionsdatei hinzu, wobei `commit-id` eine Commit-ID ist, die zur Laufzeit vom Workflow generiert wird.

Wenn Sie angeben `my-ecr-repo/image-repo:${date +%m-%d-%y-%H-%m-%s}`, fügt die Aktion der Aufgabendefinitionsdatei `my-ecr-repo/image-repo:date +%m-%d-%y-%H-%m-%s` hinzu, wobei der URI einer Amazon Elastic Container Registry (ECR) `my-ecr-repo` und `date +%m-%d-%y-%H-%m-%s` ein Zeitstempel im Format ist, das zur Laufzeit durch den Workflow `month-day-year-hour-minute-second` generiert wird.

Weitere Informationen zum `image` Feld finden Sie unter [Image](#) im Entwicklerhandbuch für Amazon Elastic Container Service. Weitere Informationen zu Variablen finden Sie unter [Arbeiten mit Variablen](#).

Entsprechende Benutzeroberfläche: Registerkarte/Image-Name der Konfiguration

## environment-variables

(*ECSRenderTaskDefinition*/Configuration/environment-variables)

(Erforderlich)

Geben Sie Umgebungsvariablen an, die die Render-Amazon-ECS-Aufgabendefinitionsaktion Ihrer Aufgabendefinitionsdatei hinzufügen soll. Die Aktion fügt die Variablen zum `environment` Feld `containerDefinitions`, in Ihrer Aufgabendefinitionsdatei hinzu. Wenn in der Datei bereits Variablen vorhanden sind, überschreibt die Aktion die Werte vorhandener Variablen und fügt alle neuen Variablen hinzu. Weitere Informationen zu Amazon-ECS-Umgebungsvariablen finden Sie unter [Angeben von Umgebungsvariablen](#) im Amazon Elastic Container Service-Entwicklerhandbuch.

Entsprechende Benutzeroberfläche: Registerkarte/Umgebungsvariablen Konfiguration – optional

## Outputs

(*ECSRenderTaskDefinition*/Outputs)

(Erforderlich)

Definiert die Daten, die von der Aktion während einer Workflow-Ausführung ausgegeben werden.

Entsprechende Benutzeroberfläche: Registerkarte „Ausgaben“

## Artifacts

(*ECSRenderTaskDefinition*/Outputs/Artifacts)

(Erforderlich)

Geben Sie die Artefakte an, die von der Aktion generiert wurden. Sie können diese Artefakte als Eingabe in anderen Aktionen referenzieren.

Weitere Informationen zu Artefakten, einschließlich Beispielen, finden Sie unter [Mit Artefakten arbeiten](#).

Entsprechende Benutzeroberfläche: Registerkarte „Outputs“/„Artifacts“

## Name

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Name)

(Erforderlich)

Geben Sie den Namen des Artefakts an, das die aktualisierte Aufgabendefinitionsdatei enthalten soll. Der Standardwert ist `MyTaskDefinitionArtifact`. Anschließend müssen Sie dieses Artefakt als Eingabe in die Aktion `Deploy to Amazon ECS` angeben. Informationen zum Hinzufügen dieses Artefakts als Eingabe zur Aktion `Deploy to Amazon ECS` finden Sie unter [Beispiel für einen Arbeitsablauf](#).

Entsprechende Benutzeroberfläche: Registerkarte/Artifacts/Name der Ausgaben

## Files

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Files)

(Erforderlich)

Geben Sie die Dateien an, die in das Artefakt aufgenommen werden sollen. Sie müssen angeben, `task-definition-*` damit die aktualisierte Aufgabendefinitionsdatei, die mit `beginnttask-definition-`, enthalten ist.

Entsprechende Benutzeroberfläche: Registerkarte/Artifacts/Files für Ausgaben

## Variables

(*ECSRenderTaskDefinition*/Outputs/Variables)

(Erforderlich)

Geben Sie den Namen einer Variablen an, die von der Renderaktion festgelegt werden soll. Die Render-Aktion legt den Wert dieser Variablen auf den Namen der aktualisierten Aufgabendefinitionsdatei fest (z. B. `task-definition-random-string.json`). Sie müssen diese Variable dann in der Eigenschaft `Deploy to Amazon ECS-Aktion Task Definition` (visueller Editor) oder `task-definition` (yaml-Editor) angeben. Informationen zum Hinzufügen dieser Variable zur Aktion `Deploy to Amazon ECS` finden Sie unter [Beispiel für einen Arbeitsablauf](#).


Standard: `task-definition`

Entsprechende Benutzeroberfläche: Ausgabe-Registerkarte/Variablen/Name-Feld

## Kontingente für Workflows in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Limits für Workflows in Amazon beschrieben CodeCatalyst.

Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

|   |   |
|---|---|
| Maximale Anzahl von Workflows pro Space   | 800   |
| Maximale Größe der Workflow-Definitionsdatei  | 256 KB  |
| Maximale Anzahl von Workflow-Dateien, die in einem einzelnen Quellereignis verarbeitet werden                         | 50  |
| Maximale Anzahl von Dateien, die in einem Einzelquellereignis verarbeitet wurden                                      | 4.000   |
| Maximale Anzahl aktiver Flotten pro Space   | 10  |
| Maximale Anzahl aktiver Recheninstanzen pro Flotte  | 20  |
| Maximale Anzahl von Eingabeartefakten pro Aktion  | 10  |
| Maximale Anzahl von Ausgabeartefakten pro Aktion  | 10  |
| Maximale Gesamtgröße der Ausgabevariablen einer einzelnen Aktion  | 120 KB  |
| Maximale Länge eines Ausgangsvariablenwerts   | 500 Zeichen oder mehr, abhängig von der Aktion, die den Wert ausgibt.   |
|   | <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note</b><br/>Werte können gekürzt werden, wenn sie das Limit der Aktion überschreiten.</p> </div> |
| Maximale Anzahl von Tagen für die Aufbewahrung von Artefakten, die während einer Workflow-Ausführung generiert wurden | 30  |

|   |   |
|---|---|
| Maximale Anzahl von Berichten pro Aktion  | 50  |
| Maximale Anzahl von Testfällen pro Testbericht                                      | 20 000  |
| Maximale Anzahl von Dateien pro Codeabdeckungsbericht                               | 20 000  |
| Maximale Anzahl von Ergebnissen der Analyse der Softwarezusammensetzung pro Bericht | 20 000  |
| Maximale Anzahl von Dateien pro statischem Analysebericht                           | 20 000  |
| Maximale Anzahl gleichzeitiger Workflow-Ausführungen pro Speicherplatz              | 100   |
| Maximale Anzahl von Aktionen pro Workflow   | 50  |
| Maximale Anzahl gleichzeitig ausgeführter Aktionen pro Workflow                     | 50  |
| Maximale Anzahl gleichzeitig ausgeführter Aktionen pro Space                        | 200   |
| Maximale Dauer, für die eine Aktion ausgeführt werden kann                          | Für die Build- und Testaktionen beträgt das Timeout 8 Stunden.<br><br>Für alle anderen Aktionen beträgt das Timeout 1 Stunde. |
| Maximale Anzahl von Umgebungen, die einem AWS-Konto pro Bereich zugeordnet sind     | 5,000   |
| Maximale Anzahl von Geheimnissen pro Aktion   | 5   |
| Maximale Anzahl von Geheimnissen pro Feld   | 500 000   |

# Probleme in CodeCatalyst

In können Sie Funktionen CodeCatalyst, Bugs und alle anderen Arbeiten, die mit Ihrem Projekt verbunden sind, überwachen. Jede Arbeit wird in einem eigenen Datensatz gespeichert, der als Problem bezeichnet wird. Sie können ein Problem in kleinere Ziele unterteilen, indem Sie ihm eine Checkliste mit Aufgaben hinzufügen. Jedes Problem kann eine Beschreibung, einen Verantwortlichen, einen Status und andere Eigenschaften haben, nach denen Sie suchen, diese gruppieren und filtern können. Sie können Ihre Probleme in den Standardansichten anzeigen oder Ihre eigenen Ansichten mit benutzerdefinierter Filterung, Sortierung oder Gruppierung erstellen. Weitere Informationen zu Problemkonzepten finden Sie unter [Probleme, Konzepte](#). Informationen zum Erstellen Ihrer ersten Ausgabe finden Sie unter [Ein Problem erstellen in CodeCatalyst](#).

Hier ist ein möglicher Arbeitsablauf für ein Team, das Probleme verwendet:

Jorge Souza ist ein Entwickler, der an einem Projekt arbeitet. Er und seine Projektkollegen Li Juan, Mateo Jackson und Wang Xiulan arbeiten zusammen, um festzustellen, welche Arbeiten erledigt werden müssen. Jeden Tag halten er und seine Entwicklerkollegen ein Synchronisationstreffen ab, das von Wang Xiulan geleitet wird. Sie rufen das Board auf, indem sie zur Ansicht des Boards in einer der Team-Ansichten navigieren. Durch die Erstellung von Ansichten können Benutzer und Teams Filter, Gruppierungen und Sortierungen von Problemen speichern, um Probleme, die ihren angegebenen Kriterien entsprechen, auf einfache Weise anzuzeigen. Ihre Ansicht enthält die Probleme, die nach dem Verantwortlichen gruppiert und nach Priorität sortiert sind, sodass die wichtigsten Probleme und der Status der Probleme für jeden Entwickler angezeigt werden. Wenn Jorge Aufgaben zugewiesen werden, plant er seine Arbeit, indem er für jede Aufgabe ein Problem erstellt. Bei der Erstellung von Problemen kann Jorge den entsprechenden Status, die Priorität und die Schätzung des Arbeitsaufwands auswählen. Bei größeren Problemen fügt Jorge der Ausgabe Aufgaben hinzu, um die Arbeit in kleinere Aufgaben aufzuteilen. Jorge erstellt seine Probleme mit dem Status „Entwurf“, z. B. als Backlog, da er nicht vorhat, sofort mit ihnen zu beginnen. Probleme im Entwurfsstatus werden in der Entwurfsansicht angezeigt, wo sie geplant und priorisiert werden müssen. Sobald Jorge bereit ist, mit der Arbeit zu beginnen, verschiebt er das entsprechende Problem auf das Board, indem er seinen Status auf einen Status in einer anderen Kategorie aktualisiert (Nicht gestartet, Gestartet oder Abgeschlossen). Während der Bearbeitung jeder Aufgabe kann das Team nach Titel, Status, Beauftragtem, Bezeichnung, Priorität und Einschätzung filtern, um ein bestimmtes Problem oder ähnliche Probleme zu finden, die dem angegebenen Parameter entsprechen. Mithilfe des Boards können Jorge und sein Team die Anzahl der für jedes Problem erledigten Aufgaben sehen und den day-to-day Fortschritt verfolgen, indem sie jedes Problem von einem Status zum nächsten ziehen, bis die Aufgabe abgeschlossen ist. Im Laufe des Projekts häufen

sich abgeschlossene Probleme mit dem Status Abgeschlossen an. Wang Xiulan beschließt, sie aus der Ansicht zu entfernen, indem er sie mithilfe der Schnellarchivierungsschaltfläche archiviert, sodass sich die Entwickler auf die Probleme konzentrieren können, die mit aktuellen und bevorstehenden Arbeiten zusammenhängen.

Bei der Planung ihrer Arbeit wählen die Entwickler, die an dem Projekt arbeiten, Sortieren nach und Gruppieren nach, um die Probleme zu finden, die sie aus dem Backlog auf das Board verschieben möchten. Sie können sich dafür entscheiden, Probleme auf der Grundlage der Kundenanfragen mit der höchsten Priorität zum Board hinzuzufügen. Daher gruppieren sie das Board nach dem Label Kundenanfrage und sortieren es nach Priorität. Sie könnten auch nach Schätzungen sortieren, um sicherzustellen, dass sie ein gewisses Arbeitsvolumen übernehmen, das sie bewältigen können. Der Projektmanager, Saanvi Sarkar, überprüft und bearbeitet regelmäßig den Rückstand, um sicherzustellen, dass die Priorität die Bedeutung der einzelnen Probleme für den Projekterfolg genau widerspiegelt.

## Themen

- [Probleme, Konzepte](#)
- [Ein Problem erstellen in CodeCatalyst](#)
- [Bearbeitung und gemeinsame Bearbeitung von Problemen in CodeCatalyst](#)
- [Probleme finden und anzeigen](#)
- [Probleme beim Exportieren](#)
- [Einstellungen für Probleme konfigurieren](#)
- [Kontingente für Ausgaben in CodeCatalyst](#)

## Probleme, Konzepte

Das Erstellen eines Problems ist eine schnelle und effiziente Möglichkeit, die im Rahmen eines Projekts geleistete Arbeit nachzuverfolgen. Sie können Probleme verwenden, um in täglichen Synchronisierungsgesprächen über die Arbeit zu sprechen, Aufgaben zu priorisieren und vieles mehr.

Diese Seite enthält eine Liste von Konzepten, die Ihnen helfen, Probleme in effektiv zu nutzen. CodeCatalyst



## Aktive Probleme

Aktive Probleme sind Probleme, bei denen es sich nicht um Probleme handelt, die sich nicht im Status Entwurf befinden oder nicht archiviert sind. Mit anderen Worten, aktive Probleme sind Probleme mit einem Status in einer der folgenden Statuskategorien: Nicht gestartet, Gestartet und Abgeschlossen. Weitere Informationen zu Status und Statuskategorien finden Sie unter [Status und Statuskategorien](#).

Sie können alle aktiven Probleme in Ihrem Projekt in der Standardansicht Aktive Probleme anzeigen.

## Archivierte Probleme

Ein archiviertes Problem ist ein Problem, das für Ihr Projekt nicht mehr relevant ist. Sie können beispielsweise [ein Problem archivieren](#), wenn es abgeschlossen ist und Sie es nicht mehr in der Spalte Erledigt sehen müssen, oder wenn es versehentlich erstellt wurde. Archivierte Probleme können bei Bedarf dearchiviert werden.

## Beauftragter

Der Beauftragte ist die Person, der das Problem zugewiesen ist. Wenn die Person nicht in der Liste erscheint, wenn Sie nach ihr suchen, wurde sie Ihrem Projekt nicht hinzugefügt. Informationen zum Hinzufügen von Personen finden Sie unter [Einen Benutzer zu Ihrem Projekt einladen](#). Informationen zur Aktivierung mehrerer Personen für ein Problem finden Sie unter [Aktivieren oder Deaktivieren mehrerer Beauftragter](#). Probleme mit mehreren Beauftragten werden auf deinem Board mit unterschiedlich farbigen Avataren angezeigt, die jeweils für einen der Verantwortlichen stehen.

## Benutzerdefinierte Felder

Mit benutzerdefinierten Feldern kannst du verschiedene Attribute eines Problems an deine Bedürfnisse anpassen, um Probleme innerhalb eines Projekts nachzuverfolgen und zu verwalten. Sie können beispielsweise ein Feld für Roadmapping, ein bestimmtes Fälligkeitsdatum oder ein Anforderungsfeld hinzufügen.

## Schätzung

In der agilen Entwicklung wird die Schätzung als Story Points bezeichnet. Verwenden Sie die Schätzung, um neben der Mehrdeutigkeit und Komplexität des Problems auch den Arbeitsaufwand einzubeziehen. Verwenden Sie höhere Schätzungen für Probleme mit größerem Risiko, größeren

Schwierigkeiten und Unbekanntes. Sie können den Schätzungstyp für Ihr Projekt ändern. Weitere Informationen zu den Schätzungstypen und deren Konfiguration finden Sie unter [Konfiguration der Problemaufwandsschätzung](#)

## Problem

Ein Problem ist ein Datensatz, in dem die Arbeit im Zusammenhang mit Ihrem Projekt aufgezeichnet wird. Sie können ein Problem für ein Feature, eine Aufgabe, einen Fehler oder jede andere Arbeit im Zusammenhang mit Ihrem Projekt erstellen. Wenn du agile Entwicklung verwendest, kann ein Problem auch ein Epos oder eine Benutzergeschichte beschreiben.

## Label (Bezeichnung)

Das Label wird verwendet, um Probleme zu gruppieren, zu sortieren und zu filtern. Sie können einen neuen Labelnamen eingeben oder eines der Labels aus der ausgefüllten Liste auswählen. Diese Liste enthält die zuletzt im Projekt verwendeten Labels. Ein Problem kann mehrere Labels haben, und ein Label kann von einem Problem entfernt werden. Informationen zum Anpassen von Bezeichnungen finden Sie unter [Labels](#).

## Priorität

Priorität bezieht sich auf den Grad der Wichtigkeit des Problems. Es gibt vier Optionen: Niedrig, Mittel, Hoch und Keine Priorität.

## Status und Statuskategorien

Der Status gibt den aktuellen Status des Problems an und wird verwendet, um den Fortschritt eines Problems während seines gesamten Lebenszyklus, von der Entstehung bis zum Abschluss, schnell zu überprüfen. Alle Probleme müssen einen Status haben, und jeder Status gehört zu einer Statuskategorie. Statuskategorien helfen Ihnen dabei, Ihren Status zu organisieren und die Standardansichten der Probleme aufzufüllen.

Es gibt fünf Standardstatus und vier Statuskategorien in CodeCatalyst. Sie können andere Status erstellen, aber Sie können keine anderen Statuskategorien erstellen. Die folgende Liste enthält die Standardstatus und ihre Statuskategorien in Klammern: Backlog (Entwurf), Zu erledigen (nicht gestartet), In Bearbeitung (Gestartet), In Bearbeitung (Gestartet) und Erledigt (Abgeschlossen).

Weitere Informationen zum Arbeiten mit Status finden Sie unter [Statuses](#)

## Aufgaben

Aufgaben können zu Vorgängen hinzugefügt werden, um die Bearbeitung dieser Ausgabe besser aufzuschlüsseln und zu organisieren. Sie können einer Ausgabe bei der Erstellung Aufgaben hinzufügen oder Aufgaben zu einer bestehenden Ausgabe hinzufügen. Wenn Sie sich ein Problem ansehen, können Sie die Aufgaben neu anordnen, entfernen oder als erledigt markieren.

## Ansichten

Probleme in Ihrem CodeCatalyst Projekt werden in Ansichten angezeigt. Bei den Ansichten kann es sich entweder um Rasteransichten handeln, in denen Probleme im Listenformat angezeigt werden, oder um Board-Ansichten, in denen Probleme als Kacheln in Spalten angezeigt werden, die nach Problemstatus geordnet sind. Es gibt vier Standardansichten, und Sie können [Ihre eigenen Ansichten mit benutzerdefinierter Gruppierung, Filterung und Sortierung erstellen](#). Die folgende Liste enthält Details zu den vier Standardansichten.

- Die Entwurfsansicht ist eine Rasteransicht, in der Probleme angezeigt werden, an denen derzeit nicht gearbeitet wird. Jedes Problem, das mit einem Status in der Kategorie Entwurfsstatus erstellt wurde, wird in dieser Ansicht angezeigt. Diese Ansicht kann von Teams verwendet werden, um zu sehen, welche Probleme noch definiert sind oder darauf warten, zugewiesen und bearbeitet zu werden.
- Die Ansicht „Aktive Probleme“ ist eine Übersicht über alle Probleme, an denen gerade gearbeitet wird. Jedes Problem mit einem Status in den Statuskategorien Nicht gestartet, Gestartet oder Abgeschlossen wird in dieser Ansicht angezeigt.
- Die Ansicht Alle Probleme ist eine Rasteransicht, in der alle Probleme im Projekt angezeigt werden, sowohl Entwürfe als auch aktive Probleme.
- In der Ansicht Archiviert werden alle archivierten Probleme angezeigt.

## Ein Problem erstellen in CodeCatalyst


Die generativen KI-Funktionen in Amazon CodeCatalyst befinden sich in der Vorschauversion und können sich ändern. Sie sind nur in der Region USA West (Oregon) verfügbar. Der Zugang zu generativen KI-Funktionen variiert je nach Stufe. Weitere Informationen finden Sie unter [-Preisgestaltung](#).

Entwicklungsteams erstellen Probleme, um ihre Arbeit nachzuverfolgen und zu verwalten. Sie können je nach Bedarf Probleme innerhalb eines Projekts erstellen. Sie könnten beispielsweise ein Problem erstellen, um die Aktualisierung einer Variablen in Ihrem Code nachzuverfolgen. Sie können anderen Benutzern im Projekt Probleme zuweisen, Labels verwenden, um Ihre Arbeit nachzuverfolgen, und vieles mehr.

Folgen Sie diesen Anweisungen, um ein Problem in zu erstellen CodeCatalyst.

Um ein Problem zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Projekt, in dem Sie ein Problem erstellen möchten.
3. Wählen Sie auf der Projekt-Startseite die Option Problem erstellen aus. Wählen Sie alternativ im Navigationsbereich Probleme aus.
4. Wählen Sie Problem erstellen aus.

 Note

Sie können Probleme auch direkt hinzufügen, wenn Sie eine Rasteransicht verwenden.

5. Geben Sie einen Titel für das Problem ein.
6. (Optional) Geben Sie eine Beschreibung ein. Sie können Markdown verwenden, um Formatierungen hinzuzufügen.
7. (Optional) Wählen Sie einen Status, eine Priorität und eine Schätzung für das Problem aus. Wenn die Einstellung für die Schätzung des Projekts auf Schätzungen ausblenden gesetzt ist, wird kein Feld „Schätzung“ angezeigt.
8. (Optional) Fügen Sie dem Problem Aufgaben hinzu. Aufgaben können verwendet werden, um die Arbeit an einer Ausgabe in kleinere Aufgaben zu unterteilen. Um eine Aufgabe hinzuzufügen, wählen Sie + Aufgaben hinzufügen. Geben Sie dann den Aufgabennamen in das Textfeld ein und drücken Sie die Eingabetaste. Nachdem Sie Aufgaben hinzugefügt haben, können Sie sie als erledigt markieren, indem Sie das Kontrollkästchen aktivieren, oder sie neu anordnen, indem Sie die Aufgabe auswählen und von der linken Seite der Checkbox ziehen.
9. (Optional) Fügen Sie ein vorhandenes Label hinzu oder erstellen Sie ein neues Label und fügen Sie es hinzu, indem Sie + Label hinzufügen wählen.

- a. Um ein vorhandenes Label hinzuzufügen, wählen Sie das Label aus der Liste aus. Sie können einen Suchbegriff in das Feld eingeben, um alle Labels im Projekt zu durchsuchen, die diesen Begriff enthalten.
  - b. Um ein neues Label zu erstellen und es hinzuzufügen, geben Sie den Namen des Labels, das Sie erstellen möchten, in das Suchfeld ein und drücken Sie die Eingabetaste.
10. (Optional) Fügen Sie einen Beauftragten hinzu, indem Sie + Bevollmächtigten hinzufügen wählen. Sie können sich schnell selbst als Beauftragten hinzufügen, indem Sie + Mich hinzufügen wählen.

 Tip

Sie können Amazon Q ein Problem zuweisen, damit Amazon Q versucht, das Problem zu lösen. Weitere Informationen finden Sie unter [Tutorial: Verwenden CodeCatalyst generativer KI-Funktionen zur Beschleunigung Ihrer Entwicklungsarbeit](#).

Diese Funktionalität setzt voraus, dass generative KI-Funktionen für den Bereich aktiviert sind. Weitere Informationen finden Sie unter [Generative KI-Funktionen verwalten](#).

11. (Optional) Fügen Sie ein vorhandenes benutzerdefiniertes Feld hinzu oder erstellen Sie ein neues benutzerdefiniertes Feld. Probleme können mehrere benutzerdefinierte Felder haben.
- a. Um ein vorhandenes benutzerdefiniertes Feld hinzuzufügen, wählen Sie das benutzerdefinierte Feld aus der Liste aus. Sie können einen Suchbegriff in das Feld eingeben, um alle benutzerdefinierten Felder im Projekt zu durchsuchen, die diesen Begriff enthalten.
  - b. Um ein neues benutzerdefiniertes Feld zu erstellen und hinzuzufügen, geben Sie den Namen des benutzerdefinierten Felds, das Sie erstellen möchten, in das Suchfeld ein und drücken Sie die Eingabetaste. Wählen Sie dann den Typ des benutzerdefinierten Felds aus, das Sie erstellen möchten, und legen Sie einen Wert fest.
12. Wählen Sie „Problem erstellen“. In der unteren rechten Ecke wird eine Benachrichtigung angezeigt: Wenn das Problem erfolgreich erstellt wurde, wird eine Bestätigungsmeldung angezeigt, dass das Problem erfolgreich erstellt wurde. Wenn das Problem nicht erfolgreich erstellt wurde, wird eine Fehlermeldung mit dem Grund für den Fehler angezeigt. Sie können dann „Erneut versuchen“ wählen, um das Problem zu bearbeiten und erneut zu erstellen, oder „Verwerfen“ wählen, um das Problem zu verwerfen. Bei beiden Optionen wird die Benachrichtigung verworfen.

**Note**

Du kannst einen Pull-Request nicht mit einem Issue verknüpfen, wenn du ihn erstellst. Du kannst ihn jedoch [bearbeiten, nachdem du ihn](#) erstellt hast, um Links zu Pull Requests hinzuzufügen.

## Bewährte Methoden bei der Erstellung und Bearbeitung von Problemen, die Amazon Q zugewiesen wurden


Wenn Sie Probleme erstellen, bleiben einige davon manchmal bestehen. Die Ursachen dafür können komplex und unterschiedlich sein. Manchmal liegt es daran, dass nicht klar ist, wer daran arbeiten soll. In anderen Fällen erfordert das Problem Recherchen oder Fachwissen in einem bestimmten Teil der Codebasis, und die besten Kandidaten für die Arbeit sind mit anderen Problemen beschäftigt. Oft gibt es andere dringende Aufgaben, die zuerst erledigt werden müssen. Eine oder alle dieser Ursachen können zu Problemen führen, an denen nicht gearbeitet wird. CodeCatalyst beinhaltet die Integration mit einem generativen KI-Assistenten namens Amazon Q, der ein Problem anhand seines Titels und seiner Beschreibung analysieren kann. Wenn Sie das Problem Amazon Q zuweisen, wird Amazon Q versuchen, einen Lösungsentwurf zu erstellen, den Sie bewerten können. Dies kann Ihnen und Ihrem Team helfen, sich auf Probleme zu konzentrieren und diese zu optimieren, während Amazon Q an einer Lösung für Probleme arbeitet, für die Sie keine Ressourcen haben, um sie sofort zu lösen.

**Note**

Bereitgestellt von Amazon Bedrock: AWS implementiert [automatisierte Missbrauchserkennung](#). Da die Funktion „Probleme an Amazon Q zuweisen“ auf Amazon Bedrock basiert, können Benutzer die in Amazon Bedrock implementierten Kontrollen voll ausschöpfen, um Sicherheit und den verantwortungsvollen Umgang mit künstlicher Intelligenz (KI) durchzusetzen.

Amazon Q schneidet bei einfachen und unkomplizierten Problemen am besten ab. Die besten Ergebnisse erzielen Sie, wenn Sie klar und deutlich erklären, was Sie tun möchten. Im Folgenden finden Sie einige bewährte Methoden, die Ihnen bei der Erstellung von Problemen helfen sollen, die für Amazon Q optimiert wurden.

- Halten Sie es einfach. Amazon Q eignet sich am besten für einfache Codeänderungen und Korrekturen, die im Titel und in der Beschreibung des Problems erklärt werden können. Weisen Sie Problemen keine vagen Titel oder übermäßig blumige oder widersprüchliche Beschreibungen zu.
- Seien Sie spezifisch. Je mehr Informationen Sie über die genauen Änderungen bereitstellen können, die zur Lösung des Problems erforderlich sind, desto wahrscheinlicher ist es, dass Amazon Q eine Lösung entwickeln kann, die das Problem löst. Geben Sie nach Möglichkeit spezifische Details an, z. B. den Namen der APIs, die Sie ändern möchten, Methoden, die Sie aktualisieren möchten, Tests, die geändert werden müssen, und alle anderen Details, die Ihnen einfallen.
- Stellen Sie sicher, dass Sie alle Details im Titel und in der Beschreibung des Problems enthalten haben, bevor Sie es Amazon Q zuweisen. Sie können den Titel oder die Beschreibung eines Problems nicht ändern, nachdem Sie es Amazon Q zugewiesen haben. Stellen Sie daher sicher, dass Sie über alle für eine Ausgabe erforderlichen Informationen verfügen, bevor Sie sie Amazon Q zuweisen.
- Weisen Sie nur Probleme, für die Codeänderungen erforderlich sind, in einem einzigen Quell-Repository zu. Amazon Q kann nur mit Code in einem einzigen Quell-Repository in arbeiten CodeCatalyst. Verknüpfte Repositories werden nicht unterstützt. Stellen Sie sicher, dass für das Problem nur Änderungen in einem einzigen Quell-Repository erforderlich sind, bevor Sie das Problem Amazon Q zuweisen.
- Verwenden Sie die von Amazon Q vorgeschlagene Standardeinstellung, um jeden Schritt zu genehmigen. Standardmäßig benötigt Amazon Q für jeden Schritt Ihre Zustimmung. Auf diese Weise können Sie mit Amazon Q nicht nur in Kommentaren zu dem Problem, sondern auch zu allen Pull-Requests, die es erstellt, interagieren. Dies bietet ein interaktiveres Erlebnis mit Amazon Q, das Ihnen hilft, den Ansatz anzupassen und den Code zu verfeinern, der zur Lösung des Problems erstellt wird.

 Note

Amazon Q reagiert nicht auf einzelne Kommentare zu Issues oder Pull Requests, überprüft sie jedoch, wenn es darum gebeten wird, seinen Ansatz zu überdenken oder eine Überarbeitung zu erstellen.

- Prüfen Sie den von Amazon Q vorgeschlagenen Ansatz immer sorgfältig. Sobald Sie den Ansatz genehmigt haben, beginnt Amazon Q mit der Generierung von Code, der auf diesem Ansatz basiert. Vergewissern Sie sich, dass der Ansatz korrekt erscheint und alle Informationen enthält, die Sie erwarten, bevor Sie Amazon Q anweisen, fortzufahren.

- Stellen Sie sicher, dass Amazon Q nur dann an Workflows arbeiten darf, wenn Sie nicht über bestehende Workflows verfügen, die diese möglicherweise bereitstellen, bevor sie überprüft wurden. In Ihrem Projekt sind möglicherweise Workflows so konfiguriert, dass sie Läufe bei Pull-Request-Ereignissen starten. Wenn ja, kann jede von Amazon Q erstellte Pull-Anfrage, die das Erstellen oder Aktualisieren von Workflow-YAML beinhaltet, eine Ausführung der in der Pull-Anfrage enthaltenen Workflows starten. Als bewährte Methode sollten Sie Amazon Q nicht erlauben, an Workflow-Dateien zu arbeiten, es sei denn, Sie sind sich sicher, dass es in Ihrem Projekt keine Workflows gibt, die diese Workflows automatisch ausführen, bevor Sie den erstellten Pull-Request überprüfen und genehmigen.

Weitere Informationen finden Sie unter [Tutorial: Verwenden CodeCatalyst generativer KI-Funktionen zur Beschleunigung Ihrer Entwicklungsarbeit](#) und [Generative KI-Funktionen verwalten](#).

## Bearbeitung und gemeinsame Bearbeitung von Problemen in CodeCatalyst

### Inhalt

- [Ein Problem bearbeiten](#)
- [Arbeiten mit Anhängen](#)
- [Aufgaben zu bestimmten Themen verwalten](#)
- [Ein Problem als blockiert oder entsperrt markieren](#)
- [Kommentare hinzufügen, bearbeiten oder löschen](#)
  - [Verwenden von Erwähnungen in einem Kommentar](#)
- [Ein Problem wird bearbeitet](#)
  - [Zwischen dem Backlog und dem Board](#)
  - [Verfolge ein Problem auf dem Board durch die einzelnen Lebenszyklusphasen](#)
  - [Probleme zwischen Gruppen verschieben](#)
- [Ein Problem archivieren](#)

### Ein Problem bearbeiten

Gehen Sie wie folgt vor, um den Titel, die Beschreibung, den Status, den Verantwortlichen, die Priorität, die Schätzung oder die Bezeichnung eines Problems zu bearbeiten.



## Um ein Problem zu bearbeiten

1. Wählen Sie das Problem aus, das Sie bearbeiten möchten, um die ProblemDetails anzuzeigen. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
2. Um den Titel der Ausgabe zu bearbeiten, wählen Sie den Titel aus, geben Sie einen neuen Titel ein und drücken Sie die Eingabetaste.
3. Um die Beschreibung zu bearbeiten, wählen Sie die Beschreibung aus, geben Sie eine neue Beschreibung ein und drücken Sie die Eingabetaste. Sie können Markdown verwenden, um Formatierungen hinzuzufügen.
4. Unter Aufgaben können Sie die Aufgaben für das Problem anzeigen und verwalten. Weitere Informationen finden Sie unter [Aufgaben zu bestimmten Themen verwalten](#).
5. Um den Status, die Schätzung oder die Priorität zu bearbeiten, wählen Sie eine Option aus den jeweiligen Drop-down-Menüs aus.
6. In Labels können Sie ein vorhandenes Label hinzufügen, ein neues Label erstellen oder ein Label entfernen.
  - a. Um ein vorhandenes Label hinzuzufügen, wählen Sie + Label hinzufügen und wählen Sie das Label aus der Liste aus. Sie können einen Suchbegriff in das Feld eingeben, um alle Labels im Projekt zu durchsuchen, die diesen Begriff enthalten.
  - b. Um ein neues Label zu erstellen und hinzuzufügen, wählen Sie + Label hinzufügen, geben Sie den Namen des Labels, das Sie erstellen möchten, in das Suchfeld ein und drücken Sie die Eingabetaste.
  - c. Um ein Etikett zu entfernen, wählen Sie das X-Symbol neben dem Etikett, das Sie entfernen möchten. Wenn Sie ein Label aus allen Ausgaben entfernen, wird das Label im Bereich Unbenutzte Labels im Bereich Labels der Ausgabeeinstellungen angezeigt. Wenn Sie Filter verwenden oder Labels zu einem Problem hinzufügen, werden unbenutzte Labels am Ende der Bezeichnungsliste angezeigt. In den Problemeinstellungen findest du eine Übersicht aller Labels (verwendet und unbenutzt) und der Probleme, die diese haben.
7. Um ein Problem zuzuweisen, wähle im Bereich Verantwortlicher die Option + Beauftragten hinzufügen aus, suche dann nach dem Verantwortlichen und wähle ihn aus der Liste aus. Du kannst + Füge mich hinzu wählen, um dich schnell selbst als Beauftragten hinzuzufügen.
8. Unter Anlagen können Sie Anlagen hinzufügen, herunterladen oder entfernen. Weitere Informationen finden Sie unter [Arbeiten mit Anhängen](#).

9. Um einen Pull Request zu verknüpfen, wähle Pull Request verknüpfen und wähle dann entweder einen Pull Request aus der Liste aus oder gib dessen URL oder ID ein. Um die Verknüpfung eines Pull-Requests aufzuheben, wähle das Symbol zum Aufheben der Verknüpfung.

 Tip

Nachdem du einen Link zu einem Pull Request zu einem Issue hinzugefügt hast, kannst du schnell zu diesem Link navigieren, indem du seine ID in der Liste der verknüpften Pull Requests auswählst. Du kannst die URL eines Pull Requests verwenden, um Pull Requests zu verknüpfen, die sich in anderen Projekten als dem Issue-Board befinden, aber nur Benutzer, die Mitglieder dieses Projekts sind, können diesen Pull Request ansehen oder zu ihm navigieren.

10. (Optional) Fügen Sie ein vorhandenes benutzerdefiniertes Feld hinzu und legen Sie es fest, erstellen Sie ein neues benutzerdefiniertes Feld oder entfernen Sie ein benutzerdefiniertes Feld. Probleme können mehrere benutzerdefinierte Felder haben.
  - a. Um ein vorhandenes benutzerdefiniertes Feld hinzuzufügen, wählen Sie das benutzerdefinierte Feld aus der Liste aus. Sie können einen Suchbegriff in das Feld eingeben, um alle benutzerdefinierten Felder im Projekt zu durchsuchen, die diesen Begriff enthalten.
  - b. Um ein neues benutzerdefiniertes Feld zu erstellen und hinzuzufügen, geben Sie den Namen des benutzerdefinierten Felds, das Sie erstellen möchten, in das Suchfeld ein und drücken Sie die Eingabetaste. Wählen Sie dann den Typ des benutzerdefinierten Felds aus, das Sie erstellen möchten, und legen Sie einen Wert fest.
  - c. Um ein benutzerdefiniertes Feld zu entfernen, wählen Sie das X-Symbol neben dem benutzerdefinierten Feld, das Sie entfernen möchten. Wenn Sie ein benutzerdefiniertes Feld aus allen Ausgaben entfernen, wird das benutzerdefinierte Feld gelöscht und Sie werden es beim Filtern nicht mehr sehen.

## Arbeiten mit Anhängen

Sie können Anlagen zu Problemen hinzufügen, CodeCatalyst um den Zugriff auf zugehörige Dateien zu erleichtern. Gehen Sie wie folgt vor, um Anlagen für ein Problem zu verwalten.

Die Größe der Anlagen, die zu Ausgaben hinzugefügt werden, wird auf die Speicherkontingente Ihres Speicherplatzes angerechnet. Informationen zum Anzeigen und Verwalten von Anhängen für Ihr Projekt finden Sie unter [Anlagen anzeigen und verwalten](#).

**⚠ Important**

Anhänge zu Problemen werden von Amazon nicht gescannt oder analysiert CodeCatalyst. Jeder Benutzer kann einer Ausgabe, die möglicherweise bösartigen Code oder schädlichen Inhalt enthält, einen Anhang hinzufügen. Stellen Sie sicher, dass die Benutzer die bewährten Methoden kennen, wenn es um die Verwaltung von Anhängen und den Schutz vor bösartigem Code, Inhalten oder Viren geht.

Um Anlagen hinzuzufügen, herunterzuladen oder zu entfernen

1. Wählen Sie das Problem aus, für das Sie Anlagen verwalten möchten. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
2. Um einen Anhang hinzuzufügen, wählen Sie Datei hochladen. Navigieren Sie im Datei-Explorer Ihres Betriebssystems zu der Datei und wählen Sie sie aus. Wählen Sie Öffnen, um sie als Anlage hinzuzufügen. Informationen zum Kontingent, wie z. B. die maximale Größe von Anhängen, finden Sie unter [Kontingente für Ausgaben in CodeCatalyst](#).

Beachten Sie die folgenden Einschränkungen in Bezug auf Dateinamen und Inhaltstypen von Anhängen:

- Die folgenden Zeichen sind in Dateinamen nicht zulässig:
  - Steuerzeichen: 0x00–0x1f und 0x80–0x9f
  - Reservierte Zeichen: /?,<,>,\,,:,\*|, und "
  - Reservierte Unix-Dateinamen: und . . .
  - Nachfolgende Punkte und Leerzeichen
  - Für Windows reservierte Dateinamen: CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9
- Der Inhaltstyp des Anhangs muss dem folgenden Medientypmuster entsprechen:

```
media-type = type "/" [tree "."] subtype ["+" suffix]* [";" parameter];
```

z. B. `text/html; charset=UTF-8`.

3. Um einen Anhang herunterzuladen, wählen Sie das Ellipsenmenü neben dem Anhang, den Sie herunterladen möchten, und wählen Sie Herunterladen.
4. Um die URL eines Anhangs zu kopieren, wählen Sie das Ellipsenmenü neben dem Anhang, dessen URL Sie kopieren möchten, und wählen Sie „URL kopieren“.
5. Um einen Anhang zu entfernen, wählen Sie das Ellipsenmenü neben dem Anhang, den Sie entfernen möchten, und wählen Sie Löschen.

## Aufgaben zu bestimmten Themen verwalten

Aufgaben können zu Vorgängen hinzugefügt werden, um die Bearbeitung des Problems weiter aufzuschlüsseln, zu organisieren und nachzuverfolgen.

Um Aufgaben zu einem Problem zu verwalten

1. Wählen Sie das Problem aus, für das Sie Aufgaben verwalten möchten. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
2. Unter Aufgaben können Sie Aufgaben für das Problem anzeigen und verwalten.
  1. Um eine Aufgabe hinzuzufügen, geben Sie den Aufgabennamen in das Textfeld ein und drücken Sie die Eingabetaste.
  2. Um eine Aufgabe als erledigt zu markieren, aktivieren Sie das Kontrollkästchen der Aufgabe.
  3. Um die Details einer Aufgabe anzuzeigen oder zu aktualisieren, wählen Sie sie aus der Liste aus.
  4. Um die Aufgaben neu anzuordnen, wählen Sie die Aufgabe aus und ziehen Sie sie von der linken Seite des Kästchens.
  5. Um eine Aufgabe zu entfernen, wählen Sie das Ellipsenmenü der Aufgabe und wählen Sie Entfernen.

## Ein Problem als blockiert oder entsperrt markieren

Wenn Sie durch etwas daran gehindert werden, an einem Problem zu arbeiten, sollten Sie es möglicherweise als blockiert markieren. Ihr Problem könnte beispielsweise blockiert werden, wenn es auf einer Änderung an einem anderen Teil Ihrer Codebasis beruht, der noch nicht zusammengeführt wurde.

Wenn du ein Problem als blockiert markierst, wird dem Problem ein rotes Etikett „Blockiert“ CodeCatalyst hinzugefügt, sodass es in deinem Backlog, Archiv oder auf deinem Board gut sichtbar ist.

Du kannst das Problem entsperren, wenn äußere Umstände behoben sind.

Um ein Problem als blockiert zu markieren

1. Öffnen Sie das Problem, das Sie als blockiert markieren möchten. Hilfe bei der Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
2. Wählen Sie Aktionen und anschließend Als blockiert markieren aus.

Um ein Problem zu entsperren

1. Öffnen Sie das Problem, das Sie entsperren möchten. Hilfe bei der Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
2. Wählen Sie „Aktionen“ und anschließend „Als entsperrt markieren“.

## Kommentare hinzufügen, bearbeiten oder löschen

Sie können einen Kommentar zu einem Problem hinterlassen. In den Kommentaren kannst du andere Space-Mitglieder, andere Projekte im Space, verwandte Probleme und Code taggen.

Um einen Kommentar zu einem Problem hinzuzufügen

1. Navigiere zu deinem Projekt.
2. Wählen Sie in der Navigationsleiste Probleme aus.
3. Wählen Sie das Problem aus, zu dem Sie den Kommentar hinzufügen möchten. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
4. Geben Sie den Kommentar in das Feld Kommentare ein. Sie können Markdown verwenden, um Formatierungen hinzuzufügen.
5. Wählen Sie Send (Senden) aus.

So bearbeiten Sie einen Kommentar

Sie können Kommentare bearbeiten, die Sie zu Problemen abgeben. Sie können nur Kommentare bearbeiten, die Sie selbst verfasst haben.

1. Navigiere zu deinem Projekt.
2. Wählen Sie in der Navigationsleiste Probleme aus.
3. Wählen Sie das Problem aus, zu dem Sie einen Kommentar bearbeiten möchten. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
4. Um einen Kommentar zu bearbeiten, suchen Sie den Kommentar, den Sie bearbeiten möchten.

 Tip

Sie können Kommentare zuerst nach dem ältesten oder dem neuesten sortieren. Es werden jeweils 10 Kommentare geladen.

5. Wählen Sie das Ellipsensymbol und dann Bearbeiten.
6. Bearbeiten Sie den Kommentar. Sie können Markdown verwenden, um Formatierungen hinzuzufügen.
7. Wählen Sie Speichern. Der Kommentar ist jetzt aktualisiert.

### So löschen Sie einen Kommentar

Sie können Kommentare löschen, die Sie zu Problemen abgegeben haben. Sie können nur Kommentare löschen, die Sie selbst verfasst haben. Wenn ein Kommentar gelöscht wird, wird Ihr Benutzername angezeigt, jedoch mit den Worten Dieser Kommentar wurde gelöscht anstelle des ursprünglichen Kommentartextes.

1. Navigiere zu deinem Projekt.
2. Wählen Sie in der Navigationsleiste Probleme aus.
3. Wählen Sie das Problem aus, bei dem Sie einen Kommentar löschen möchten. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
4. Wählen Sie das Ellipsensymbol, dann Löschen und anschließend Bestätigen aus.

### Verwenden von Erwähnungen in einem Kommentar

Sie können Space-Mitglieder, andere Projekte im Space, verwandte Probleme und Code in Kommentaren erwähnen. Dadurch wird ein Schnelllink zu dem Benutzer oder der Ressource erstellt, die Sie erwähnen.

## An @mention in einem Kommentar

1. Navigiere zu deinem Projekt.
2. Wählen Sie in der Navigationsleiste Probleme aus.
3. Wählen Sie das Problem aus, das Sie bearbeiten möchten, um die ProblemDetails anzuzeigen. Hilfe zur Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
4. Wählen Sie das Textfeld Kommentar hinzufügen aus.
5. Geben Sie ein `@user_name`, um einen anderen Benutzer zu erwähnen.
6. Geben Sie ein `@project_name`, um ein Projekt zu erwähnen.
7. Geben Sie ein `@issue_name` oder `@issue_number` um ein anderes Problem zu erwähnen.
8. Geben Sie `@file_name` Text ein, um bestimmte Dateien oder Code in einem Quell-Repository zu erwähnen.

### Note

Eine Liste der fünf wichtigsten Elemente (Benutzer, Quell-Repositorys, Projekte usw.) mit den Begriffen, die zu Ihnen passen, `@mention` wird während der Eingabe ausgefüllt.

9. Wählen Sie den gewünschten Artikel aus, den Sie erwähnen möchten. Der Pfad, der anzeigt, wo sich der Artikel befindet, wird in das Kommentartextfeld eingefügt.
10. Beenden Sie Ihren Kommentar und wählen Sie Senden.

## Ein Problem wird bearbeitet

Jedes Problem hat einen Lebenszyklus. CodeCatalystIn: Probleme beginnen in der Regel als Entwurf im Backlog. Wenn mit der Bearbeitung dieses Problems begonnen werden soll, wird es in eine andere Statuskategorie verschoben und durchläuft verschiedene Status, bis es abgeschlossen ist. Anschließend wird es archiviert. Sie können ein Problem auf folgende Weise verschieben oder während seines gesamten Lebenszyklus weiterbearbeiten:

- Du kannst ein Problem zwischen dem Backlog und dem Board verschieben.
- Du kannst ein Problem, das gerade bearbeitet wird, durch verschiedene Fertigstellungsphasen verschieben.
- Sie können ein Problem archivieren, das abgeschlossen ist.

## Zwischen dem Backlog und dem Board

Sie können ein Problem aus dem Backlog in das Board verschieben, sobald Sie mit der Bearbeitung des Problems beginnen. Du kannst ein Problem auch wieder in den Backlog verschieben, wenn die Arbeit verschoben wird.

Um ein Problem zwischen dem Backlog und dem Board zu verschieben

1. Navigiere zu deinem Projekt.
2. Wählen Sie im Navigationsbereich Probleme aus. Die Standardansicht ist das Board.
3. Um ein Problem vom Board in den Backlog zu verschieben:
  - a. Wähle das Problem aus, das du verschieben möchtest. Hilfe bei der Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
  - b. Wählen Sie im Dropdownmenü Status die Option Backlog aus.
4. Um ein Problem aus dem Backlog auf das Board zu verschieben:
  - a. Um zum Backlog zu navigieren, wähle Board und dann Backlog.
  - b. Wähle das Problem aus, das du verschieben möchtest. Hilfe bei der Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).
  - c. Wähle Zum Board hinzufügen oder wähle einen anderen Status als Backlog.

## Verfolge ein Problem auf dem Board durch die einzelnen Lebenszyklusphasen

Du kannst ein Problem innerhalb eines Forums durch verschiedene Statusstufen verschieben, bis es abgeschlossen ist.

Um ein Problem innerhalb des Forums zu verschieben

1. Wählen Sie im Navigationsbereich Probleme aus. Die Standardansicht ist das Board.
2. Führen Sie eine der folgenden Aktionen aus:
  - Ziehen Sie ein Problem per Drag-and-Drop in einen anderen Status.
  - Wählen Sie ein Problem und dann im Dropdownmenü Status einen Status aus.
  - Wählen Sie ein Problem aus und wählen Sie dann „Gehe zu: **nächstem** Status“.



Informationen zur Archivierung eines Problems finden Sie unter [Ein Problem archivieren](#).

## Probleme zwischen Gruppen verschieben

Sie können [Probleme in den Ansichten „Alle Probleme“ und „Forum“ nach verschiedenen Parametern gruppieren](#). Wenn die Probleme gruppiert sind, können Sie Probleme von einer Gruppe in eine andere verschieben. Wenn Sie ein Problem von einer Gruppe in eine andere verschieben, wird das Feld, in dem die Probleme gruppiert sind, automatisch entsprechend der Zielgruppe bearbeitet.

Nehmen wir als Beispielszenario an, es gibt ein Unternehmen CodeCatalyst, das Probleme zwei Personen, Wang Xiulan und Saanvi Sarkar, zugewiesen hat. Der Vorstand ist nach Gruppen gruppiert, und es gibt zwei Gruppen, eine für jeden Beauftragten. Wenn ein Problem von der Wang Xiulan-Gruppe in die Saanvi Sarkar-Gruppe verschoben wird, wird der Verantwortliche für das Problem auf Saanvi Sarkar aktualisiert.

## Ein Problem archivieren

### Note

Probleme werden nicht innerhalb eines Projekts gelöscht, sondern archiviert. Um Probleme zu löschen, müssen Sie das Projekt löschen.

Sie können ein Problem archivieren, wenn es in Ihrem Projekt nicht mehr benötigt wird. Wenn Sie ein Problem archivieren, CodeCatalyst wird es aus allen Ansichten entfernt, in denen archivierte Probleme herausgefiltert werden. Archivierte Probleme können in der Standardansicht Archivierte Probleme angezeigt werden. Dort können sie bei Bedarf aus der Archivierung entfernt werden.

Sie archivieren ein Problem, wenn:

- Sie haben das Problem abgeschlossen und benötigen es nicht mehr in der Spalte Erledigt.
- Sie haben nicht vor, daran zu arbeiten.
- Sie haben es irrtümlich erstellt.
- Sie haben die maximale Anzahl aktiver Probleme erreicht.

Um ein Problem zu archivieren

1. Öffnen Sie die Ausgabe, die Sie archivieren möchten. Hilfe bei der Suche nach Ihrem Problem finden Sie unter [Probleme finden und anzeigen](#).

2. Wählen Sie „Aktionen“ und anschließend „In Archiv verschieben“.
3. (Optional) Um schnell mehrere Ausgaben mit dem Status Abgeschlossen zu archivieren, klicken Sie auf der Pinnwand auf die vertikale Ellipse über dem Status Abgeschlossen und wählen Sie Probleme archivieren.

Um die Archivierung einer Ausgabe aufzuheben

1. Öffnen Sie die Ausgabe, deren Archivierung Sie aufheben möchten. Sie können sich eine Liste der archivierten Probleme anzeigen lassen, indem Sie die Ansicht Archivierte Probleme im Dropdownmenü „Problemansicht“ öffnen. Hilfe bei der Suche nach Ihrem Problem finden Sie unter. [Probleme finden und anzeigen](#)
2. Wählen Sie Unarchivieren aus.

## Probleme finden und anzeigen

In den folgenden Abschnitten wird beschrieben, wie Sie innerhalb eines CodeCatalyst Projekts effektiv nach Problemen suchen und diese anzeigen können.

Probleme in Ihrem CodeCatalyst Projekt werden in Ansichten angezeigt. Bei den Ansichten kann es sich entweder um Rasteransichten handeln, in denen Probleme im Listenformat angezeigt werden, oder um Board-Ansichten, in denen Probleme als Kacheln in Spalten angezeigt werden, die nach Problemstatus geordnet sind. Es gibt vier Standardansichten, und Sie können [Ihre eigenen Ansichten mit benutzerdefinierter Gruppierung, Filterung und Sortierung erstellen](#). Die folgende Liste enthält Details zu den vier Standardansichten.

- Die Entwurfsansicht ist eine Rasteransicht, in der Probleme angezeigt werden, an denen derzeit nicht gearbeitet wird. Jedes Problem, das mit einem Status in der Kategorie Entwurfsstatus erstellt wurde, wird in dieser Ansicht angezeigt. Diese Ansicht kann von Teams verwendet werden, um zu sehen, welche Probleme noch definiert sind oder darauf warten, zugewiesen und bearbeitet zu werden.
- Die Ansicht „Aktive Probleme“ ist eine Übersicht über alle Probleme, an denen gerade gearbeitet wird. Jedes Problem mit einem Status in den Statuskategorien Nicht gestartet, Gestartet oder Abgeschlossen wird in dieser Ansicht angezeigt.
- Die Ansicht Alle Probleme ist eine Rasteransicht, in der alle Probleme im Projekt angezeigt werden, sowohl Entwürfe als auch aktive Probleme.
- In der Ansicht Archiviert werden alle archivierten Probleme angezeigt.

## Sie suchen nach einem Problem

Sie können ein Problem finden, indem Sie nach bestimmten Parametern suchen. Weitere Informationen zur Verfeinerung Ihrer Suche finden Sie unter [Suche in CodeCatalyst](#).

Um nach einem Problem zu suchen

1. Navigieren Sie zu Ihrem Projekt.
2. Verwenden Sie die Suchleiste, um nach Problemen oder Informationen zu Problemen zu suchen. Sie können Abfrageparameter verwenden, um Ihre Suche zu verfeinern. Weitere Informationen finden Sie unter [Suche in CodeCatalyst](#).

## Probleme beim Sortieren

Standardmäßig CodeCatalyst werden Probleme in manueller Reihenfolge sortiert. Bei manueller Reihenfolge werden Probleme in der Reihenfolge angezeigt, in die sie von den Benutzern verschoben wurden. In manueller Reihenfolge sortierte Probleme können Sie per Drag-and-Drop verschieben, um ihre Reihenfolge zu ändern. Diese Sortieroption ist hilfreich, wenn es darum geht, den Problembestand zu korrigieren und Probleme zu priorisieren.

Die folgende Tabelle zeigt, wie Probleme sowohl in der Raster- als auch in der Board-Ansicht sortiert werden können.

| Sortieroptionen in der Rasteransicht | Sortieroptionen für die Board-Ansicht |
|--------------------------------------|---------------------------------------|
| Manuelle Bestellung                  | Manuelle Bestellung                   |
| Letzte Aktualisierung                | Letzte Aktualisierung                 |
| Priorität                            | Priorität                             |
| Schätzung                            | Schätzung                             |
| Title                                | Title                                 |
| ID                                   |                                       |
| Status                               |                                       |

| Sortieroptionen in der Rasteransicht | Sortieroptionen für die Board-Ansicht |
|--------------------------------------|---------------------------------------|
| Blocked                              |                                       |
| Benutzerdefinierte Felder            |                                       |

Gehen Sie wie folgt vor, um die Sortierung von Problemen zu ändern.

Um Probleme zu sortieren

1. Navigiere zu deinem Projekt.
2. Wählen Sie im Navigationsbereich Issues aus. Die Standardansicht ist das Board.
3. (Optional) Wählen Sie Aktive Probleme aus, um das Drop-down-Menü zum Wechseln der Vorgangsansicht zu öffnen und zu einer anderen Problemansicht zu gelangen.
4. Um eine Rasteransicht zu sortieren, gibt es zwei Optionen:
  - a. Wählen Sie die Überschrift des Felds aus, nach dem Sie sortieren möchten. Bei der Auswahl der Kopfzeile wird zwischen aufsteigender und absteigender Reihenfolge gewechselt.
  - b. Wählen Sie das Dropdownmenü Sortieren nach und wählen Sie einen Parameter aus, nach dem sortiert werden soll. Die Probleme werden in aufsteigender Reihenfolge sortiert.
5. Um eine Board-Ansicht zu sortieren, wähle das Drop-down-Menü Sortieren nach und wähle einen Parameter aus, nach dem sortiert werden soll. Die Probleme werden in aufsteigender Reihenfolge sortiert.

## Probleme gruppieren

Die Gruppierung wird verwendet, um Probleme auf dem Board nach mehreren Parametern zu organisieren, wie z. B. dem Verantwortlichen, der Bezeichnung und der Priorität.

Um Probleme zu gruppieren

1. Navigiere zu deinem Projekt.
2. Wählen Sie im Navigationsbereich Issues aus. Die Standardansicht ist das Board.
3. (Optional) Wählen Sie Aktive Probleme aus, um das Drop-down-Menü zum Wechseln der Vorgangsansicht zu öffnen und zu einer anderen Problemansicht zu gelangen.

4. Wählen Sie „Gruppe“.
5. Wählen Sie unter Gruppieren nach einen Parameter aus, nach dem gruppiert werden soll:
  - Wenn Sie „Beauftragter“ oder „Priorität“ wählen, wählen Sie die Gruppenreihenfolge.
  - Wenn Sie „Label“ wählen, wählen Sie die Labels und dann „Gruppenreihenfolge“ aus.
6. (Optional) Wählen Sie den Schalter Leere Gruppen anzeigen, um Gruppen ein- oder auszublenden, denen derzeit keine Probleme zugewiesen sind.
7. Die Ansicht wird aktualisiert, wenn Sie Ihre Auswahl treffen. Ein Problem tritt nur in der Gruppe auf, die den konfigurierten Parametern entspricht.

## Probleme beim Filtern

Verwenden Sie die Filterung, um Probleme zu finden, die einen bestimmten Namen, eine bestimmte Priorität, ein bestimmtes Label, benutzerdefinierte Felder oder einen bestimmten Verantwortlichen enthalten.

Um Probleme zu filtern

1. Navigiere zu deinem Projekt.
2. Wählen Sie im Navigationsbereich Issues aus.
3. (Optional) Wählen Sie Aktive Probleme aus, um das Dropdownmenü zum Wechseln der Problemansicht zu öffnen und zu einer anderen Problemansicht zu wechseln.

### Note

Um nach einer Zeichenfolge im Problemnamen oder in der Beschreibung zu filtern, geben Sie die Zeichenfolge in die Problemsuchleiste ein.

4. Wählen Sie „Filter“ und anschließend „+ Filter hinzufügen“.
5. Wählen Sie die Parameter aus, nach denen gefiltert werden soll. Sie können mehrere Filter und Parameter auswählen. Sie können Filter so konfigurieren, dass Probleme angezeigt werden, die jedem Filter oder einem einzelnen Filter entsprechen, indem Sie und oder oder auswählen. Die Ansicht wird aktualisiert und zeigt nun die Probleme an, die dem Filter entsprechen.

## Eine Problemansicht erstellen

Sie können [Ansichten](#) erstellen, um schnell Probleme anzuzeigen, die einem bestimmten Filtersatz entsprechen. Auf diese Weise können Sie Zeit sparen und Probleme, nach denen Sie zuvor gefiltert, gruppiert oder sortiert haben, schnell anzeigen.

Um eine Problemansicht zu erstellen

1. Wählen Sie im Navigationsbereich Probleme aus.
2. (Optional) Je nach Anwendungsfall möchten Sie möglicherweise eine Ansicht aus einer vorhandenen Ansicht erstellen. Um zu einer anderen Ansicht zu wechseln, wählen Sie Aktive Probleme aus, um das Dropdownmenü zum Umschalten der Problemansicht zu öffnen, und wählen Sie die Ansicht aus.
3. (Optional) Konfigurieren Sie Filter, Gruppierung und Sortierung, bevor Sie Ihre Ansicht erstellen. Sie können diese beim Erstellen einer Ansicht hinzufügen. Wenn Sie dies jedoch zuvor tun, können Sie eine Vorschau der in der Ansicht angezeigten Elemente anzeigen, bevor Sie sie erstellen.
4. Öffnen Sie in der Kopfleiste das Dropdownmenü zum Umschalten zwischen den Themenansichten. Um eine Board-Ansicht zu erstellen, in der Probleme je nach Status in Spalten angezeigt werden, wähle das + in der Board-Spalte. Um eine Rasteransicht zu erstellen, in der Probleme in einer Liste angezeigt werden, wähle das + in der Rasterspalte. Sie können den Ansichtstyp vor der Erstellung ändern, wenn Sie es sich anders überlegen.
5. Geben Sie im Dialogfeld Ansicht erstellen einen Namen für die Ansicht ein.
6. Die Felder Filter, Probleme gruppieren nach und Probleme sortieren werden entsprechend den Einstellungen der aktuellen Ansicht gefüllt. Aktualisieren Sie sie bei Bedarf.
7. Wählen Sie Ansicht erstellen, um die Ansicht zu erstellen und zu ihr zu wechseln.

## Probleme beim Exportieren

Sie können Probleme in Ihrer aktuellen Ansicht in eine XLSX-Datei exportieren. Gehen Sie wie folgt vor, um Probleme zu exportieren.

Um Probleme zu exportieren

1. Navigieren Sie zu Ihrem Projekt.
2. Wählen Sie in der Navigationsleiste Probleme aus.

3. Wählen Sie Aktive Probleme, um das Drop-down-Menü zum Umschalten der Problemansicht zu öffnen, und navigieren Sie zu der Ansicht, die die Probleme enthält, die Sie exportieren möchten. Nur die in der Ansicht angezeigten Probleme werden exportiert.
4. Wählen Sie das Ellipsenmenü und dann Nach Excel exportieren.
5. Die XLSX-Datei wird heruntergeladen. Standardmäßig trägt sie den Namen des Projekts und das Datum, an dem der Export abgeschlossen wurde.

## Einstellungen für Probleme konfigurieren

In den folgenden Themen wird detailliert beschrieben, wie Sie Einstellungen für Probleme in konfigurieren CodeCatalyst.

Themen

- [Aktivieren oder Deaktivieren mehrerer Beauftragter](#)
- [Konfiguration der Problemaufwandsschätzung](#)
- [Statuses](#)
- [Labels](#)
- [Benutzerdefinierte Felder](#)
- [Anlagen anzeigen und verwalten](#)

## Aktivieren oder Deaktivieren mehrerer Beauftragter

Gehen Sie wie folgt vor, um die Einstellung für mehrere Verantwortliche für Probleme in zu konfigurieren. CodeCatalyst

Um mehrere Beauftragte zu aktivieren oder zu deaktivieren

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie Aktive Probleme aus, um das Drop-down-Menü „Issues View Switcher“ zu öffnen, und wählen Sie Einstellungen aus.
3. Schalten Sie unter „Verantwortlicher“ im Bereich „Grundeinstellungen“ die Anzeige um, damit mehrere Beauftragte demselben Problem zugewiesen werden können. Ein Problem kann bis zu 10 Beauftragte haben. Wenn Sie diese Option nicht aktivieren, können Sie einem Problem nur einen Verantwortlichen zuweisen.

## Konfiguration der Problemaufwandsschätzung

Gehen Sie wie folgt vor, um die Einstellung für Aufwandsschätzungen bei Problemen in zu konfigurieren. CodeCatalyst

So konfigurieren Sie die Aufwandsschätzung für Probleme

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie Aktive Probleme aus, um das Drop-down-Menü „Issues View Switcher“ zu öffnen, und wählen Sie Einstellungen aus.
3. Wählen Sie unter Schätzung im Bereich Grundeinstellungen aus, wie die Schätzwerte angezeigt werden sollen. Es stehen folgende Schätztypen zur Verfügung: T-Shirt-Größen, Fibonacci-Sequenzierung oder Schätzungen ausblenden. Wenn der Schätzungstyp aktualisiert wird, gehen keine Daten verloren, und der Schätzwert aller Probleme wird automatisch umgerechnet. Die Konversionszuweisung ist in der folgenden Tabelle dargestellt.

| Größe des T-Shirts | Fibonacci-Sequenz |
|--------------------|-------------------|
| XS                 | 1                 |
| XS                 | 2                 |
| S                  | 3                 |
| M                  | 5                 |
| L                  | 8                 |
| XL                 | 13                |

## Statuses

Du kannst deinem Board benutzerdefinierte Status hinzufügen. Jeder benutzerdefinierte Status muss zu einer der folgenden Kategorien gehören: Entwurf, Nicht gestartet, Gestartet oder Abgeschlossen. Statuskategorien werden verwendet, um den Status zu organisieren und Standardansichten aufzufüllen. Weitere Informationen zu Status und Statuskategorien finden [Status und Statuskategorien](#) Sie unter. Weitere Informationen zu Ansichten finden Sie unter. [Probleme finden und anzeigen](#)



## Um einen Status zu erstellen

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie Aktive Probleme aus, um das Drop-down-Menü „Issues View Switcher“ zu öffnen, und wählen Sie Einstellungen aus.
3. Wählen Sie unter Status das Plusymbol neben der Kategorie aus, in der sich der Status befinden soll.
4. Geben Sie dem Status einen Namen und wählen Sie dann das Häkchensymbol.

### Note

Wählen Sie das X-Symbol, um das Hinzufügen eines Status abubrechen.

Der benutzerdefinierte Status ist jetzt auf deinem Board sichtbar und wird als Option angezeigt, wenn du ein Problem erstellst.

## So bearbeiten Sie einen Status

1. Wähle im Navigationsbereich Probleme aus.
2. Wählen Sie Aktive Probleme aus, um das Drop-down-Menü „Issues View Switcher“ zu öffnen, und wählen Sie Einstellungen aus.
3. Wählen Sie unter Status das Bearbeitungssymbol neben dem Status, den Sie bearbeiten oder ändern möchten.
4. Bearbeiten Sie den Status und wählen Sie dann das Häkchensymbol.

Der bearbeitete Status ist jetzt auf deinem Board sichtbar.

## Um einen Status zu verschieben

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie das Ellipsensymbol und dann Einstellungen aus.
3. Wählen Sie unter Status einen Status aus, den Sie verschieben möchten.
4. Ziehen Sie den Status per Drag-and-Drop an die gewünschte Position.

**Note**

Sie können einen Status nur innerhalb der dafür vorgesehenen Kategorie verschieben.

Die Status sind jetzt auf deinem Board neu angeordnet.

## Um einen Status zu deaktivieren

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie Aktive Probleme aus, um das Drop-down-Menü „Issues View Switcher“ zu öffnen, und wählen Sie Einstellungen aus.
3. Wählen Sie unter Status einen Status aus, den Sie deaktivieren möchten.
4. Wählen Sie bei dem Status, den Sie deaktivieren möchten, den Schalter für den Status aus. Der Status ist jetzt ausgegraut.

**Note**

Der Status „Deaktiviert“ wird auf dem Board angezeigt, bis alle Probleme aus dem Forum entfernt wurden. Probleme können nicht zu einem deaktivierten Status hinzugefügt werden.

5. Um einen deaktivierten Status wieder zu aktivieren, klicken Sie auf den entsprechenden Schalter. Der Status ist nicht mehr ausgegraut.

**Note**

In jeder Kategorie muss mindestens ein aktiver Status vorhanden sein. Wenn es in der Kategorie nur einen Status gibt, können Sie ihn nicht deaktivieren.

## Labels

Sie können Labels für Probleme anpassen. Dazu gehören das Bearbeiten des Labels und das Ändern der Farbe. Labels können Ihnen helfen, Ihre Arbeit zu kategorisieren und zu organisieren.

## Ein Etikett erstellen

In erstellen Sie Labels CodeCatalyst, indem Sie sie entweder hinzufügen, wenn Sie eine neue Ausgabe erstellen, oder wenn Sie eine bestehende Ausgabe bearbeiten. Weitere Informationen finden Sie unter [Ein Problem erstellen in CodeCatalyst](#) und [Bearbeitung und gemeinsame Bearbeitung von Problemen in CodeCatalyst](#).

## Ein Label bearbeiten

Gehen Sie wie folgt vor, um den Namen oder die Farbe eines vorhandenen Etiketts zu ändern.

Um ein Etikett zu bearbeiten

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie Aktive Probleme aus, um das Drop-down-Menü „Issues View Switcher“ zu öffnen, und wählen Sie Einstellungen aus.
3. Auf der Kachel „Labels“ befindet sich eine Liste der im Projekt verwendeten Labels. Wählen Sie das Bearbeitungssymbol neben dem Etikett, das Sie bearbeiten möchten. Führen Sie eine oder mehrere der folgenden Aktionen aus:
  - a. Bearbeiten Sie den Namen des Labels.
  - b. Um die Farbe zu ändern, wählen Sie das Farbrad. Verwenden Sie den Picker, um eine neue Farbe auszuwählen.
4. Um die Änderungen zu speichern, die Sie am Etikett vorgenommen haben, wählen Sie das Häkchensymbol.
5. Das geänderte Etikett ist jetzt in Ihrer Liste der verfügbaren Labels sichtbar. Sie können auch sehen, für wie viele Ausgaben dieses Label verwendet wird.

### Note

Sie können die Zahl auswählen, die neben jedem Etikett angezeigt wird, um zur Seite Alle Probleme zu navigieren und alle Ausgaben zu sehen, die dieses Label enthalten.

## Löschen eines Labels

Derzeit können Sie ein Problemlabel in nicht löschen CodeCatalyst. Wenn Sie ein Label aus allen Ausgaben entfernen, wird das Label im Bereich Unbenutzte Labels im Bereich Labels der

Problemeinstellungen angezeigt. Wenn Sie Filter verwenden oder Labels zu einem Problem hinzufügen, werden unbenutzte Labels am Ende der Bezeichnungsliste angezeigt. In den Problemeinstellungen findest du eine Übersicht aller Labels (verwendet und unbenutzt) und der Probleme, die diese haben.

## Benutzerdefinierte Felder

Sie können benutzerdefinierte Felder erstellen, um die Arbeit für Ihr Projekt zu organisieren und anzuzeigen. Benutzerdefinierte Felder werden der Liste der verfügbaren Filter in Filter hinzugefügt, sodass Sie Probleme nach benutzerdefinierten Feldern filtern können. Benutzerdefinierte Felder sind Namens- und Wertepaare. Sie filtern nach dem Namen des benutzerdefinierten Felds und dann nach dem Wert dieses benutzerdefinierten Felds.

Ein Problem kann mehrere benutzerdefinierte Felder haben.

### Ein benutzerdefiniertes Feld erstellen

In erstellen Sie benutzerdefinierte Felder CodeCatalyst, indem Sie sie entweder hinzufügen, wenn Sie ein Problem erstellen oder wenn Sie ein vorhandenes Problem bearbeiten. Weitere Informationen finden Sie unter [Ein Problem erstellen in CodeCatalyst](#) und [Bearbeitung und gemeinsame Bearbeitung von Problemen in CodeCatalyst](#).

### Löschen eines benutzerdefinierten Felds

Um ein benutzerdefiniertes Feld zu löschen, müssen Sie das benutzerdefinierte Feld aus jeder Ausgabe entfernen, zu der es hinzugefügt wird. Wenn ein benutzerdefiniertes Feld gelöscht wird, wird das benutzerdefinierte Feld nicht mehr im Filter angezeigt. Sie können Filter verwenden, um alle Probleme mit einem benutzerdefinierten Feld anzuzeigen und sie zu entfernen, indem Sie die Probleme bearbeiten. Weitere Informationen finden Sie unter [Probleme finden und anzeigen](#) und [Ein Problem bearbeiten](#)

## Anlagen anzeigen und verwalten

In den Problemeinstellungen können Sie eine Tabelle mit allen Anlagen anzeigen, die zu Problemen in Ihrem Projekt hinzugefügt wurden. Diese Tabelle enthält Details zu jedem Anhang, darunter Informationen wie den Inhaltstyp, den Zeitpunkt, zu dem er hinzugefügt wurde, das Problem, dem er hinzugefügt wurde, seinen Status und die Dateigröße.

Anhand dieser Tabelle können Sie mühelos große Dateianhänge identifizieren, die abgeschlossen oder archiviert wurden, um sie zu entfernen, um Speicherplatz freizugeben.

**⚠ Important**

Anhänge zu Problemen werden von Amazon nicht gescannt oder analysiert CodeCatalyst. Jeder Benutzer kann einer Ausgabe, die möglicherweise bösartigen Code oder schädlichen Inhalt enthält, einen Anhang hinzufügen. Stellen Sie sicher, dass die Benutzer die bewährten Methoden kennen, wenn es um die Verwaltung von Anhängen und den Schutz vor bösartigem Code, Inhalten oder Viren geht.

So können Sie alle Problemanhänge in einem Projekt anzeigen und verwalten

1. Wählen Sie im Navigationsbereich Probleme aus.
2. Wählen Sie das Ellipsensymbol und dann Einstellungen aus.
3. Wählen Sie die Registerkarte „Anlagen“.

## Kontingente für Ausgaben in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Limits für Probleme bei Amazon beschrieben CodeCatalyst. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

| Ressource                                      | Standardkontingent   |
|--|--|
| Aktive Probleme                                | Maximal 1.000 pro Projekt.   |
| Größe des Anhangs                              | Maximal 500 MB pro Anhang.<br><br>Der maximale Gesamtspeicher für Anhänge wird durch die Gesamtspeicherbeschränkungen für Ihren Speicherplatz beeinflusst. Weitere Informationen finden Sie unter <a href="#">-Preisgestaltung</a> . |
| Gesamtzahl der Probleme (aktiv und archiviert) | Maximal 100.000 pro Projekt.   |
| Gespeicherte Ansichten                         | Maximal 50 gespeicherte Problemansichten pro Projekt.  |

| Ressource  | Standardkontingent                  |
|--|-------------------------------------|
| Anzahl der Pull-Requests, die du mit einem Issue verknüpfen kannst | Maximal 50 Pull-Requests pro Issue. |
| Status (pro Projekt)   | Maximal 50 pro Projekt.             |
| Status (pro Ausgabe)   | Maximal 50 pro Ausgabe.             |
| Labels (pro Projekt)   | Maximal 200 pro Projekt.            |
| Etiketten (pro Ausgabe)  | Maximal 50 pro Ausgabe.             |
| Benutzerdefinierte Felder (pro Ausgabe)                            | Maximal 50 pro Ausgabe.             |
| Beauftragte  | Maximal 10 pro Ausgabe.             |
| Kommentare   | Maximal 1.000 pro Ausgabe.          |
| Aufgaben   | Maximal 100 pro Ausgabe.            |

# Identität, Berechtigungen und Zugriff in CodeCatalyst

Wenn Sie sich zum ersten Mal bei Amazon CodeCatalyst anmelden, erstellen Sie eine AWS Builder-ID. AWS Builder-IDs existieren nicht in AWS Identity and Access Management. Der Benutzername, den Sie bei Ihrer ersten Anmeldung wählen, wird zu Ihrer eindeutigen Benutzer-ID für Ihre Identität.

CodeCatalystIn können Sie sich auf zwei Arten zum ersten Mal anmelden:

- Im Rahmen der Schaffung eines Raums.
- Im Rahmen der Annahme einer Einladung zu einem Projekt oder Raum in CodeCatalyst.

Die Rolle oder Rollen, die mit Ihrer Identität verknüpft sind, bestimmen die Aktionen, die Sie ausführen können CodeCatalyst. Projektrollen wie Projektadministrator und Mitwirkender sind projektspezifisch, sodass Sie in einem Projekt eine Rolle und in einem anderen Projekt eine andere Rolle haben können. Wenn Sie einen Bereich erstellen, CodeCatalyst wird Ihnen automatisch die Rolle Space-Administrator zugewiesen. Wenn Benutzer Einladungen zu einem Projekt annehmen, CodeCatalyst fügt diese Identitäten dem Bereich hinzu und weist ihnen die Rolle Eingeschränkter Zugriff zu. Wenn Sie Benutzer zu Projekten einladen, wählen Sie die Rolle aus, die sie im Projekt haben sollen. Dadurch wird bestimmt, welche Aktionen sie innerhalb des Projekts ausführen können und welche nicht. Die meisten Benutzer, die an einem Projekt arbeiten, benötigen nur die Rolle Mitwirkender, um ihre Aufgaben auszuführen. Weitere Informationen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

Zusätzlich zu einer Projektrolle benötigen Benutzer in einem Projekt ein persönliches Zugriffstoken (PAT), um auf Quell-Repositorys für ein Projekt zuzugreifen, wenn sie Git-Clients oder integrierte Entwicklungsumgebungen (IDEs) verwenden. Projektmitglieder können dieses PAT mit Anwendungen von Drittanbietern als anwendungsspezifisches Passwort verwenden, das mit ihrer Identität verknüpft ist. CodeCatalyst Wenn Sie beispielsweise ein Quell-Repository auf einen lokalen Computer klonen, müssen Sie neben Ihrem CodeCatalyst Benutzernamen auch eine PAT angeben.

Sie können den Zugriff zwischen CodeCatalyst und AWS Ressourcen konfigurieren, indem Sie eine [Servicerolle](#) verwenden, um Aktionen wie den Zugriff auf AWS CloudFormation Stacks und Ressourcen auszuführen, wenn Sie Aktionen in Workflows bereitstellen. Sie müssen den Zugriff zwischen CodeCatalyst und AWS Ressourcen konfigurieren, damit die Workflow-Aktionen, die in den Projektvorlagen enthalten sind, ausgeführt werden können.

Themen

- [Mit Rollen in Amazon arbeiten CodeCatalyst](#)
- [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#)
- [Multi-Faktor-Authentifizierung \(MFA\) bei Amazon CodeCatalyst](#)
- [Sicherheit bei Amazon CodeCatalyst](#)
- [Überwachung bei Amazon CodeCatalyst](#)
- [Kontingente für Identität, Erlaubnis und Zugriff in CodeCatalyst](#)
- [Fehlerbehebung](#)

## Mit Rollen in Amazon arbeiten CodeCatalyst

In Amazon CodeCatalyst können Sie Benutzern Rollen sowohl auf Projekt- als auch auf Bereichsebene zuweisen. In einem Projekt gibt eine Rolle an, was ein Benutzer in einem Projekt mit den Ressourcen für dieses Projekt tun darf. Benutzer erhalten Mitgliedschaft in einem Bereich, wenn sie einem Projekt beitreten. Sie können Benutzer als Administratoren eines Bereichs hinzufügen oder entfernen. Die Rolle des Space-Administrators verfügt über die umfassendsten Berechtigungen aller Rollen in CodeCatalyst. Es hat sich bewährt, Benutzern die engsten Berechtigungen zuzuweisen, die für die Ausführung ihrer Aufgaben erforderlich sind.

Sie können Benutzern in dem Bereich Rollen zuweisen. Sie können Benutzern auch Rollen in den Projekten zuweisen, in denen sie Mitglied sind. Jeder Benutzer kann nur eine Rolle in einem Projekt oder Bereich haben, aber Benutzer können in jedem Projekt und Bereich unterschiedliche Rollen haben. Beispielsweise kann ein Benutzer in einem Projekt die Rolle Projektadministrator und in einem anderen Projekt die Rolle Mitwirkender haben.

### Themen

- [Rollentypen](#)
- [Für jede Rolle verfügbare Berechtigungen](#)
- [Benutzerrollen anzeigen und ändern](#)

## Rollentypen

Für einen Bereich sind drei Rollen verfügbar:

- Bereichsadministrator



- Hauptbenutzer
- Eingeschränkter Zugriff

Benutzern, die eine Einladung zu einem Projekt annehmen, wird in dem Bereich, der das Projekt enthält, automatisch die Rolle Eingeschränkter Zugriff zugewiesen.

Mitgliedern eines Projekts stehen vier Rollen zur Verfügung:

- Projektadministrator
- Beitragender
- Gutachter
- Nur lesen

Wenn Sie einem Projekt einen Benutzer hinzufügen, CodeCatalyst wird ihm automatisch die Rolle Eingeschränkter Zugriff zugewiesen. Wenn Sie einen Benutzer aus allen Projekten entfernen, CodeCatalyst wird diesem Benutzer automatisch die Rolle mit beschränktem Zugriff entzogen.

## Rolle „Space-Administrator“

Die Rolle des Space-Administrators ist die mächtigste Rolle in CodeCatalyst. Weisen Sie die Rolle Space-Administrator nur Benutzern zu, die jeden Aspekt eines Bereichs verwalten müssen, da diese Rolle über alle erforderlichen Berechtigungen verfügt. CodeCatalyst Benutzer mit der Rolle Space-Administrator sind die einzigen Benutzer, die andere Benutzer zur Space-Administratorrolle hinzufügen oder daraus entfernen und den Space löschen können.

Wenn Sie einen Bereich erstellen, CodeCatalyst wird Ihnen automatisch die Rolle Space-Administrator zugewiesen. Als bewährte Methode empfehlen wir, dass Sie diese Rolle mindestens einem anderen Benutzer hinzufügen, der in dieser Rolle agieren kann, falls der ursprüngliche Space-Ersteller nicht verfügbar ist.

## Die Rolle des Hauptbenutzers

Die Poweruser-Rolle ist die zweitwichtigste Rolle in CodeCatalyst Spaces, hat aber keinen Zugriff auf Projekte in einem Space. Sie ist für Benutzer konzipiert, die in der Lage sein müssen, Projekte in einem Bereich zu erstellen und sie bei der Verwaltung der Benutzer und Ressourcen für den Bereich zu unterstützen. Weisen Sie Benutzern die Rolle Poweruser zu, die Teamleiter oder Manager sind und im Rahmen ihrer Arbeit Projekte erstellen und Benutzer im Bereich verwalten müssen.

## Rolle mit beschränktem Zugriff

Die Rolle mit beschränktem Zugriff ist die Rolle, die die meisten Benutzer in CodeCatalyst Spaces haben werden. Diese Rolle wird Benutzern automatisch zugewiesen, wenn sie eine Einladung zu einem Projekt in einem Space annehmen. Sie bietet die eingeschränkten Berechtigungen, die sie benötigen, um in dem Bereich zu arbeiten, der das Projekt enthält. Weisen Sie Benutzern, die Sie direkt in den Bereich einladen, die Rolle „Eingeschränkter Zugriff“ zu, es sei denn, sie müssen für ihre Arbeit einen Teil des Bereichs verwalten.

## Rolle als Projektadministrator

Die Rolle des Projektadministrators ist die mächtigste Rolle in einem CodeCatalyst Projekt. Weisen Sie diese Rolle nur Benutzern zu, die alle Aspekte eines Projekts verwalten müssen, einschließlich der Bearbeitung von Projekteinstellungen, der Verwaltung von Projektberechtigungen und dem Löschen von Projekten.

Projektrollen haben keine Berechtigungen auf Bereichsebene. Daher können Benutzer mit der Rolle Projektadministrator keine zusätzlichen Projekte erstellen. Nur Benutzer mit der Rolle Space-Administrator oder Poweruser können Projekte erstellen.

### Note

Die Rolle Space-Administrator verfügt über alle Berechtigungen für CodeCatalyst.

## Rolle „Mitwirkender“

Die Rolle Mitwirkender ist für die Mehrheit der Mitglieder eines Projekts vorgesehen. CodeCatalyst Weisen Sie diese Rolle Benutzern zu, die in der Lage sein müssen, mit Code, Workflows, Problemen und Aktionen in einem Projekt zu arbeiten.

## Rolle des Prüfers

Die Rolle „Prüfer“ richtet sich an Benutzer, die in der Lage sein müssen, mit Ressourcen in einem Projekt zu interagieren, z. B. mit Pull-Requests und Issues, aber keinen Code zu erstellen und zusammenzuführen, Workflows zu erstellen oder Workflow-Ausführungen in einem CodeCatalyst Projekt zu starten oder zu beenden. Weisen Sie Benutzern die Rolle Prüfer zu, die in der Lage sein müssen, Pull-Requests zu genehmigen und zu kommentieren, Probleme zu erstellen, zu aktualisieren, zu lösen und zu kommentieren sowie Code und Workflows in einem Projekt einzusehen.

## Rolle „Nur lesen“

Die Rolle Schreibgeschützt ist für Benutzer vorgesehen, die die Ressourcen und den Status von Ressourcen einsehen müssen, aber nicht mit ihnen interagieren oder direkt zum Projekt beitragen möchten. Benutzer mit dieser Rolle können keine Ressourcen in erstellen CodeCatalyst, sie können sie jedoch anzeigen und kopieren, z. B. Repositories klonen und Dateianhänge auf einen lokalen Computer herunterladen. Weisen Sie Benutzern, die Ressourcen und den Status des Projekts einsehen, aber nicht direkt mit ihnen interagieren müssen, die Rolle Schreibgeschützt zu.

## Für jede Rolle verfügbare Berechtigungen

Die folgende Tabelle zeigt die für jede CodeCatalyst Rolle verfügbaren Berechtigungen. Verwenden Sie die Links, um zu den entsprechenden Berechtigungen zu springen.

- [Space permissions](#)
- [Extensions permissions](#)
- [Project permissions](#)
- [Source repository permissions](#)
- [Dev Environment permissions](#)
- [Package repository and package permissions](#)
- [Workflow permissions](#)
- [Issues permissions](#)
- [Custom blueprint permissions](#)
- [Notifications permissions](#)
- [Search permissions](#)






















| Berechtigung | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|--------------|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
|              |                                  |                              |                                   |                                |                        |                   |                  |





























Platzberechtigungen

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Raum schaffen   |    |    |    |    |    |    |    |
| Bearbeiten Sie die Abrechnungsdetails für den Speicherplatz |    |    |    |    |    |    |    |
| Richten Sie Single Sign-On ein und aktivieren Sie es        |    |    |    |    |    |    |    |
| Single Sign-On entfernen                                    |  |  |  |  |  |  |  |
| Aktivieren Sie generative KI-Funktionen für einen Raum      |  |  |  |  |  |  |  |










































| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Deaktivieren<br>generative KI-Funktionen für einen Raum       |    |    |    |    |    |    |    |
| Bereich löschen   |    |    |    |    |    |    |    |
| Fügen Sie der Space-Administratorrolle weitere Benutzer hinzu |   |   |   |   |   |   |   |
| Andere Benutzer aus der Space-Administratorrolle entfernen    |  |  |  |  |  |  |  |

| Berechtig ung  | Rolle eines Space-Adm inistrators | Die Rolle des Hauptbenu tzers | Rolle mit eingeschr änkttem Zugriff | Rolle als Projektad ministrat or | Rolle als Mitwirken der | Rolle des Prüfers | Rolle nur lesbar |
|--|-----------------------------------|-------------------------------|-------------------------------------|----------------------------------|-------------------------|-------------------|------------------|
| Team erstellen   |                                   |                               |                                     |                                  |                         |                   |                  |
| Team löschen   |                                   |                               |                                     |                                  |                         |                   |                  |
| Team aktualisi eren  |                                   |                               |                                     |                                  |                         |                   |                  |
| Deaktivie ren<br>Sie die Maschinen ressourc en für den Bereich |                                   |                               |                                     |                                  |                         |                   |                  |
| Aktiviere n Sie die Maschinen ressourc en für den Bereich      |                                   |                               |                                     |                                  |                         |                   |                  |
| Projekt erstellen  |                                   |                               |                                     |                                  |                         |                   |                  |

| Berechtigung                                       | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Ordnen Sie dem Bereich eine AWS-Kontoverbindung zu |    |    |    |    |    |    |    |
| Eine AWS-Kontoverbindung aktualisieren             |    |    |    |    |    |    |    |
| Trennen Sie eine AWS-Kontoverbindung vom Bereich   |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Löschen Sie eine AWS-Kontoverbindung und entfernen Sie sie aus dem Bereich |    |    |    |    |    |    |    |
| Laden Sie andere in den Bereich ein  |    |    |    |    |    |    |    |
| VPC-Verbindung erstellen   |  |  |  |  |  |  |  |
| VPC-Verbindung bearbeiten  |  |  |  |  |  |  |  |







| Berechtigung                             | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| VPC-Verbindung löschen                   |    |    |    |    |    |    |    |
| Aktivitätsprotokolle im Bereich anzeigen |    |    |    |    |    |    |    |
| AWS-Kontoverbindungen anzeigen           |    |    |    |    |    |    |    |
| Vorfälle anzeigen für CodeCatalyst       |  |  |  |  |  |  |  |
| Raum anzeigen                            |  |  |  |  |  |  |  |
| Teams anzeigen                           |  |  |  |  |  |  |  |





| Berechtigung                     | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|----------------------------------|---|---|---|---|---|---|---|
| VPC-Verbindungen anzeigen        |    |    |    |    |    |    |    |
| Berechtigungen für Erweiterungen | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
| Installieren Sie Erweiterungen   |    |    |    |    |    |    |    |
| Erweiterungen aktualisieren      |  |  |  |  |  |  |  |
| Erweiterungen löschen            |  |  |  |  |  |  |  |
| Ein GitHub Konto Connect         |  |  |  |  |  |  |  |








| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Verbindung zu einem GitHub Konto trennen                      |    |    |    |    |    |    |    |
| Eine Jira-Site Connect  |    |    |    |    |    |    |    |
| Trennen Sie die Verbindung zu einer Jira-Site                 |    |    |    |    |    |    |    |
| Konfigurationsdetails für installierte Erweiterungen anzeigen |  |  |  |  |  |  |  |
| Erweiterungen anzeigen  |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Projektberechtigungen                                    | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
| Projekteinstellungen bearbeiten                          |    |    |    |    |    |    |    |
| Deaktivieren Sie die Maschinenressourcen für das Projekt |   |   |   |   |   |   |   |
| Aktivieren Sie Maschinenressourcen für das Projekt       |  |  |  |  |  |  |  |
| Projekt löschen  |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Laden Sie Benutzer zu einem Projekt ein              |    |    |    |    |    |    |    |
| Ändern Sie die Rollen von Benutzern in einem Projekt |    |    |    |    |    |    |    |
| Benutzer aus einem Projekt entfernen                 |  |  |  |  |  |  |  |
| Team zu einem Projekt hinzufügen                     |  |  |  |  |  |  |  |

























































| Berechtigung                                  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Team aus einem Projekt entfernen              |    |    |    |    |    |    |    |
| Ändern Sie die Projekteigenschaften des Teams |    |    |    |    |    |    |    |
| Projekt ansehen                               |   |   |   |   |   |   |   |
| Projekttätigkeit anzeigen                     |  |  |  |  |  |  |  |
| Teams im Projekt anzeigen                     |  |  |  |  |  |  |  |
| Baupläne anzeigen                             |  |  |  |  |  |  |  |


| Berechtigung                            | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Berechtigungen für das Quell-Repository |   |   |   |   |   |   |   |
| Repository erstellen                    |    |    |    |    |    |    |    |
| Repository verknüpfen                   |    |    |    |    |    |    |    |
| Verknüpfung mit Repository aufheben     |  |  |  |  |  |  |  |
| Löschen Sie Repository                  |  |  |  |  |  |  |  |
| Repository-Einstellungen bearbeiten     |  |  |  |  |  |  |  |



































| Berechtigung                      | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|-----------------------------------|---|---|---|---|---|---|---|
| Repositorys anzeigen              |    |    |    |    |    |    |    |
| Repository-Einstellungen anzeigen |    |    |    |    |    |    |    |
| Repositorys klonen                |    |    |    |    |    |    |    |
| Filialen erstellen                |   |   |   |   |   |   |   |
| Erstellen Sie Regeln für Filialen |  |  |  |  |  |  |  |
| Ändern Sie den Standardzweig      |  |  |  |  |  |  |  |
| Zweige löschen                    |  |  |  |  |  |  |  |


































| Berechtigung                              | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|---|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
| Zusammenführen von Zweigen                |                                  |                              |                                   |                                |                        |                   |                  |
| Aktualisieren Sie die Regeln für Filialen |                                  |                              |                                   |                                |                        |                   |                  |
| Filialen anzeigen                         |                                  |                              |                                   |                                |                        |                   |                  |
| Regeln für Filialen anzeigen              |                                  |                              |                                   |                                |                        |                   |                  |
| Ordner erstellen                          |                                  |                              |                                   |                                |                        |                   |                  |
| Ordner löschen                            |                                  |                              |                                   |                                |                        |                   |                  |
| Ordner bearbeiten                         |                                  |                              |                                   |                                |                        |                   |                  |
| Ordner anzeigen                           |                                  |                              |                                   |                                |                        |                   |                  |

| Berechtig<br>ung   | Rolle<br>eines<br>Space-<br>Adm<br>inistrators                                      | Die Rolle<br>des<br>Hauptbenu<br>tzers  | Rolle mit<br>eingeschr<br>änktem<br>Zugriff   | Rolle als<br>Projektad<br>ministrat<br>or   | Rolle als<br>Mitwirken<br>der   | Rolle des<br>Prüfers  | Rolle nur<br>lesbar   |
|--|---|---|---|---|---|---|---|
| Dateien<br>erstellen                                       |    |    |    |    |    |    |    |
| -Dateien<br>löschen  |    |    |    |    |    |    |    |
| Dateien<br>bearbeite<br>n                                  |    |    |    |    |    |    |    |
| Dateien<br>ansehen   |    |    |    |    |    |    |    |
| Commits<br>erstellen<br>und<br>pushen                      |   |   |   |   |   |   |   |
| Commits<br>anzeigen  |  |  |  |  |  |  |  |
| Erstellen<br>Sie Pull-<br>Requests                         |  |  |  |  |  |  |  |
| Erstelle<br>Genehmigu<br>ngsregeln<br>für Pull<br>Requests |  |  |  |  |  |  |  |

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Überschreiben Sie die Merge-Anforderungen für Pull Requests |    |    |    |    |    |    |    |
| Pull-Requests aktualisieren                                 |    |    |    |    |    |    |    |
| Aktualisieren Sie die Genehmigungsregeln für Pull-Requests  |  |  |  |  |  |  |  |
| Anzeigen von Pull-Anforderungen                             |  |  |  |  |  |  |  |

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Genehmigungsregeln für Pull Requests anzeigen                 |    |    |    |    |    |    |    |
| Pull Requests schließen                                       |    |    |    |    |    |    |    |
| Genehmigen Sie Pull-Anfragen                                  |    |    |    |    |    |    |    |
| Kommentieren Sie Pull-Requests                                |  |  |  |  |  |  |  |
| Interagieren Sie mit Amazon Q in Kommentaren zu Pull-Requests |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Eine Revision für einen von Amazon Q erstellten Pull-Request erstellen |    |    |    |    |    |    |    |
| Verknüpfen Sie ein Problem mit einem Pull-Request                      |    |    |    |    |    |    |    |
| Trennen Sie die Verknüpfung eines Issues mit einer Pull-Anfrage        |  |  |  |  |  |  |  |



















| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Berechtigungen für die Entwicklungsumgebung                                   |   |   |   |   |   |   |   |
| Erstellen Sie Ihre eigene Entwicklungsumgebung                                |    |    |    |    |    |    |    |
| Stoppen Sie Ihre eigene Entwicklungsumgebung                                  |  |  |  |  |  |  |  |
| Stoppen Sie Entwicklungsumgebungen, die von anderen Benutzern erstellt wurden |  |  |  |  |  |  |  |

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Setzen Sie Ihre eigene Entwicklungsumgebung fort                                    |    |    |    |    |    |    |    |
| Sehen Sie sich Ihre eigenen Entwicklungsumgebungen an                               |    |    |    |    |    |    |    |
| Sehen Sie sich Entwicklungsumgebungen an, die von anderen Benutzern erstellt wurden |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirker | Rolle des Prüfers | Rolle nur lesbar |
|--|----------------------------------|------------------------------|-----------------------------------|--------------------------------|---------------------|-------------------|------------------|
| Bearbeiten Sie Ihre eigene Entwicklungsumgebung                                  |                                  |                              |                                   |                                |                     |                   |                  |
| Bearbeiten Sie Entwicklungsumgebungen, die von anderen Benutzern erstellt wurden |                                  |                              |                                   |                                |                     |                   |                  |
| Löschen Sie Ihre eigene Entwicklungsumgebung                                     |                                  |                              |                                   |                                |                     |                   |                  |





































| Berechtigung  | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirker | Rolle des Prüfers | Rolle nur lesbar |
|---|----------------------------------|------------------------------|-----------------------------------|--------------------------------|---------------------|-------------------|------------------|
| Löschen Sie Entwicklungsumgebungen, die von anderen Benutzern erstellt wurden |                                  |                              |                                   |                                |                     |                   |                  |
| Erstellen Sie eine Dev-Datei für Entwicklungsumgebungen                       |                                  |                              |                                   |                                |                     |                   |                  |
| Bearbeiten Sie eine Dev-Datei für Entwicklungsumgebungen                      |                                  |                              |                                   |                                |                     |                   |                  |

| Berechtigung                                       | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Löscht eine Dev-Datei für Entwicklungsumgebungen   |    |    |    |    |    |    |    |
| Eine Dev-Datei für Entwicklungsumgebungen anzeigen |    |    |    |    |    |    |    |
| Paket-Repository und Paketberechtigungen           | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
| Paket-Repository erstellen                         |  |  |  |  |  |  |  |

| Berechtigung                               | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|--|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
| Paket-Repositorys anzeigen                 |                                  |                              |                                   |                                |                        |                   |                  |
| Paket-Repository bearbeiten                |                                  |                              |                                   |                                |                        |                   |                  |
| Paket-Repository löschen                   |                                  |                              |                                   |                                |                        |                   |                  |
| Erstellen Sie ein Gateway-Paket-Repository |                                  |                              |                                   |                                |                        |                   |                  |
| Gateway-Paket-Repositorys anzeigen         |                                  |                              |                                   |                                |                        |                   |                  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Löschen Sie das Gateway-Paket-Repository                     |    |    |    |    |    |    |    |
| Fügen Sie das Upstream-Paket-Repository hinzu                |    |    |    |    |    |    |    |
| Bearbeiten Sie die Suchreihenfolge der Upstream-Repositories |  |  |  |  |  |  |  |
| Entfernen Sie das Upstream-Paket-Repository                  |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirker | Rolle des Prüfers | Rolle nur lesbar |
|--|----------------------------------|------------------------------|-----------------------------------|--------------------------------|---------------------|-------------------|------------------|
| Stellen Sie eine Connect zu einem Paket-Repository her |                                  |                              |                                   |                                |                     |                   |                  |
| Liest Pakete aus einem Paket-Repository                |                                  |                              |                                   |                                |                     |                   |                  |
| Veröffentlichen Sie Pakete in einem Paket-Repository   |                                  |                              |                                   |                                |                     |                   |                  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Pakete aus einem Upstream-Repository lesen und aufbewahren |    |    |    |    |    |    |    |
| Pakete anzeigen  |    |    |    |    |    |    |    |
| Paketversionen anzeigen                                    |  |  |  |  |  |  |  |
| Ressourcen zur Paketversion anzeigen                       |  |  |  |  |  |  |  |
| Listet die Abhängigkeiten der Paketversionen auf           |  |  |  |  |  |  |  |


















































| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Aktualisieren den Status der Paketversion              |    |    |    |    |    |    |    |
| Aktualisieren Sie die Konfiguration des Paketursprungs |    |    |    |    |    |    |    |
| Paketversion löschen                                   |  |  |  |  |  |  |  |
| Workflow-Berechtigungen                                | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
| Workflow erstellen                                     |  |  |  |  |  |  |  |
| Arbeitsablauf aktualisieren                            |  |  |  |  |  |  |  |

| Berechtigung                       | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|------------------------------------|---|---|---|---|---|---|---|
| Workflow löschen                   |    |    |    |    |    |    |    |
| Workflow starten                   |    |    |    |    |    |    |    |
| Arbeitsablauf beenden              |    |    |    |    |    |    |    |
| Workflow-Geheimnisse erstellen     |    |    |    |    |    |    |    |
| Workflow-Geheimnisse aktualisieren |  |  |  |  |  |  |  |
| Workflow-Geheimnisse löschen       |  |  |  |  |  |  |  |
| Erstellen von Umgebungen.          |  |  |  |  |  |  |  |



| Berechtigung                                     | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|--|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
| Umgebungen löschen                               |                                  |                              |                                   |                                |                        |                   |                  |
| Flotte erstellen                                 |                                  |                              |                                   |                                |                        |                   |                  |
| Flotte aktualisieren                             |                                  |                              |                                   |                                |                        |                   |                  |
| Flotte löschen                                   |                                  |                              |                                   |                                |                        |                   |                  |
| Workflow-Ressourcen in anderen Konten verwalten  |                                  |                              |                                   |                                |                        |                   |                  |
| Ordnen Sie eine VPC-Verbindung einer Umgebung zu |                                  |                              |                                   |                                |                        |                   |                  |


| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Trennen Sie eine VPC-Verbindung mit einer Umgebung                         |    |    |    |    |    |    |    |
| Ordnen Sie eine VPC-verbundene Umgebung einem Workflow zu                  |    |    |    |    |    |    |    |
| Trennen Sie die Zuordnung einer VPC-verbundenen Umgebung zu einem Workflow |  |  |  |  |  |  |  |

| Berechtigung                                 | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Verfolgen Sie einen Commit in einem Workflow |    |    |    |    |    |    |    |
| Umgebungen anzeigen                          |    |    |    |    |    |    |    |
| Aktionsprotokolle für Builds anzeigen        |    |    |    |    |    |    |    |
| Flotten anzeigen                             |  |  |  |  |  |  |  |
| Testaktionsprotokolle anzeigen               |  |  |  |  |  |  |  |
| Arbeitsablauf anzeigen                       |  |  |  |  |  |  |  |
| Workflow-Läufe anzeigen                      |  |  |  |  |  |  |  |

| Berechtigung                                | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|---|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
| Ergebnisse der Workflow-Ausführung anzeigen |                                  |                              |                                   |                                |                        |                   |                  |
| Workflow-Geheimnisse anzeigen               |                                  |                              |                                   |                                |                        |                   |                  |
| Erteilt Berechtigungen                      | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
| Problem erstellen                           |                                  |                              |                                   |                                |                        |                   |                  |
| Problem aktualisieren                       |                                  |                              |                                   |                                |                        |                   |                  |
| Probleme anzeigen                           |                                  |                              |                                   |                                |                        |                   |                  |
| Ein Problem archivieren                     |                                  |                              |                                   |                                |                        |                   |                  |

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Amazon Q ein Problem zuweisen                                       |    |    |    |    |    |    |    |
| Interagieren Sie mit Amazon Q in Kommentaren zu einem Problem       |    |    |    |    |    |    |    |
| Zuordnung von Amazon Q zu einem Problem aufheben                    |  |  |  |  |  |  |  |
| Probleme aktualisieren, die von anderen Benutzern verursacht wurden |  |  |  |  |  |  |  |

| Berechtigung                                   | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|--|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
| Kommentare zu einem Problem anzeigen           |                                  |                              |                                   |                                |                        |                   |                  |
| Einen Kommentar zu einem Problem erstellen     |                                  |                              |                                   |                                |                        |                   |                  |
| Aktualisieren einen Kommentar zu einem Problem |                                  |                              |                                   |                                |                        |                   |                  |
| Erstellen Sie ein Etikett                      |                                  |                              |                                   |                                |                        |                   |                  |
| Aktualisieren Sie ein Etikett                  |                                  |                              |                                   |                                |                        |                   |                  |
| Labels anzeigen                                |                                  |                              |                                   |                                |                        |                   |                  |

| Berechtig-<br>ung  | Rolle<br>eines<br>Space-<br>Adm<br>inistrators                                      | Die Rolle<br>des<br>Hauptbenu-<br>tzers   | Rolle mit<br>eingeschr-<br>änktem<br>Zugriff  | Rolle als<br>Projektad-<br>ministrat-<br>or   | Rolle als<br>Mitwirken-<br>der  | Rolle des<br>Prüfers  | Rolle nur<br>lesbar   |
|--|---|---|---|---|---|---|---|
| Fügen<br>Sie<br>einem<br>Problem<br>ein Label<br>hinzu                       |    |    |    |    |    |    |    |
| Ein<br>Etikett<br>von<br>einem<br>Problem<br>entfernen                       |    |    |    |    |    |    |    |
| Erstellen<br>Sie einen<br>benutzerd-<br>efinierten<br>Status für<br>Probleme |  |  |  |  |  |  |  |
| Aktualisi-<br>eren Sie<br>einen<br>benutzerd-<br>efinierten<br>Status        |  |  |  |  |  |  |  |












| Berechtigung                                      | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirkender | Rolle des Prüfers | Rolle nur lesbar |
|---|----------------------------------|------------------------------|-----------------------------------|--------------------------------|------------------------|-------------------|------------------|
| Einen benutzerdefinierten Status anzeigen         |                                  |                              |                                   |                                |                        |                   |                  |
| Einen benutzerdefinierten Status verschieben      |                                  |                              |                                   |                                |                        |                   |                  |
| Deaktivieren Sie einen benutzerdefinierten Status |                                  |                              |                                   |                                |                        |                   |                  |
| Fügen Sie einem Problem einen Anhang hinzu        |                                  |                              |                                   |                                |                        |                   |                  |



| Berechtig-<br>ung   | Rolle<br>eines<br>Space-<br>Adm<br>inistrators                                      | Die Rolle<br>des<br>Hauptbenu<br>tzers  | Rolle mit<br>eingeschr<br>änktem<br>Zugriff   | Rolle als<br>Projektad<br>ministrat<br>or   | Rolle als<br>Mitwirken<br>der   | Rolle des<br>Prüfers  | Rolle nur<br>lesbar   |
|---|---|---|---|---|---|---|---|
| Einen<br>Probleman<br>hang<br>anzeigen                              |    |    |    |    |    |    |    |
| Einen<br>Anhang<br>aus<br>einem<br>Problem<br>entfernen             |    |    |    |    |    |    |    |
| Verknüpfe<br>n Sie<br>eine Pull-<br>Anfrage<br>mit einem<br>Problem |   |   |   |   |   |   |   |
| Trennen<br>Sie eine<br>Pull-Anfr<br>age von<br>einem<br>Issue       |  |  |  |  |  |  |  |
| Verknüpfe<br>n Sie ein<br>Jira-Proj<br>ekt                          |  |  |  |  |  |  |  |

| Berechtigung   | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|--|---|---|---|---|---|---|---|
| Die Verknüpfung eines Jira-Projekts aufheben                     |    |    |    |    |    |    |    |
| Benutzerdefinierte Blueprint-Berechtigungen                      | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
| Erstellen Sie ein benutzerdefiniertes Blueprint-Projekt          |  |  |  |  |  |  |  |
| Veröffentlichen Sie einen benutzerdefinierten Vorschau-Blueprint |  |  |  |  |  |  |  |

| Berechtig<br>ung   | Rolle<br>eines<br>Space-<br>Adm<br>inistrators                                      | Die Rolle<br>des<br>Hauptbenu<br>tzers  | Rolle mit<br>eingeschr<br>änktem<br>Zugriff   | Rolle als<br>Projektad<br>ministrat<br>or   | Rolle als<br>Mitwirken<br>der   | Rolle des<br>Prüfers  | Rolle nur<br>lesbar   |
|--|---|---|---|---|---|---|---|
| Machen<br>Sie die<br>Veröffent<br>lichung<br>eines<br>benutzerd<br>efinierten<br>Vorschau-<br>Blueprint<br>s<br>rückgängi<br>g |    |    |    |    |    |    |    |
| Veröffent<br>lichen<br>Sie einen<br>benutzerd<br>efinierten<br>Blueprint   |  |  |  |  |  |  |  |
| Machen<br>Sie die<br>Veröffent<br>lichung<br>eines<br>benutzerd<br>efinierten<br>Blueprint<br>s<br>rückgängi<br>g              |  |  |  |  |  |  |  |

| Berechtigung  | Rolle eines Space-Administrators   | Die Rolle des Hauptbenutzers   | Rolle mit eingeschränktem Zugriff  | Rolle als Projektadministrator   | Rolle als Mitwirkender   | Rolle des Prüfers  | Rolle nur lesbar   |
|---|--|--|--|--|--|--|--|
| Fügen Sie einen benutzerdefinierten Blueprint zu einem Space-Blueprints-Katalog hinzu |   |   |   |   |   |   |   |
| Entfernen Sie einen benutzerdefinierten Blueprint aus einem Space-Blueprints-Katalog  |  |  |  |  |  |  |  |

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Verwalten Sie die Veröffentlichungsberechtigungen für einen benutzerdefinierten Blueprint |    |    |    |    |    |    |    |
| Verwalten Sie die Katalogversion für einen benutzerdefinierten Blueprint                  |    |    |    |    |    |    |    |
| Aktualisieren Sie einen benutzerdefinierten Blueprint                                     |  |  |  |  |  |  |  |















| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Löschen Sie eine benutzerdefinierte Blueprint-Version             |    |    |    |    |    |    |    |
| Löschen Sie einen benutzerdefinierten Blueprint                   |    |    |    |    |    |    |    |
| Wenden Sie einen benutzerdefinierten Blueprint auf ein Projekt an |  |  |  |  |  |  |  |
| Trennen Sie einen benutzerdefinierten Blueprint von einem Projekt |  |  |  |  |  |  |  |

| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirker   | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Aktualisieren Sie die Version eines angewendeten benutzerdefinierten Blueprints |    |    |    |    |    |    |    |
| Bearbeiten Sie den Alias eines benutzerdefinierten Blueprints                   |   |   |   |   |   |   |   |
| Veröffentlichen benutzerdefinierte Blueprints anzeigen                          |  |  |  |  |  |  |  |

| Berechtigung                              | Rolle eines Space-Administrators | Die Rolle des Hauptbenutzers | Rolle mit eingeschränktem Zugriff | Rolle als Projektadministrator | Rolle als Mitwirker | Rolle des Prüfers | Rolle nur lesbar |
|---|----------------------------------|------------------------------|-----------------------------------|--------------------------------|---------------------|-------------------|------------------|
| Berechtigungen für Benachrichtigungen     |                                  |                              |                                   |                                |                     |                   |                  |
| Benachrichtigungskanal konfigurieren      |                                  |                              |                                   |                                |                     |                   |                  |
| Benachrichtigungskanal entfernen          |                                  |                              |                                   |                                |                     |                   |                  |
| Benachrichtigungseinstellungen bearbeiten |                                  |                              |                                   |                                |                     |                   |                  |
| Benachrichtigungseinstellungen anzeigen   |                                  |                              |                                   |                                |                     |                   |                  |



| Berechtigung  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|---|---|---|---|---|---|---|---|
| Erhalten Sie automatisch Benachrichtigungen über CodeCatalyst Vorfälle        |  |  |  |  |  |  |  |
| Konfigurieren Sie E-Mail-Benachrichtigungen für Ihr zugeordnetes E-Mail-Konto |  |  |  |  |  |  |  |
| Suchberechtigungen  | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |

| Berechtigung                        | Rolle eines Space-Administrators  | Die Rolle des Hauptbenutzers  | Rolle mit eingeschränktem Zugriff   | Rolle als Projektadministrator  | Rolle als Mitwirkender  | Rolle des Prüfers   | Rolle nur lesbar  |
|-------------------------------------|---|---|---|---|---|---|---|
| Suchen Sie innerhalb eines Projekts |  |  |  |  |  |  |  |
| Durchsuchen den gesamten Raum       |  |  |  |  |  |  |  |

## Benutzerrollen anzeigen und ändern

Sie können die einem Benutzer zugewiesene Rolle anzeigen. Auf diese Weise können Sie besser verstehen, welche Aktionen sie in einem Projekt ergreifen können. Sie können ihre Rolle auch ändern, wenn sie zusätzliche Berechtigungen benötigen.

Um die Rolle eines Benutzers in einem Projekt anzuzeigen

1. Navigieren Sie zu dem Projekt, in dem Sie die Rollen der einzelnen Projektmitglieder anzeigen möchten.

### Tip

In der oberen Navigationsleiste können Sie auswählen, welches Projekt angezeigt werden soll.

2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Auf der Registerkarte Mitglieder wird die Rolle für jedes Projektmitglied unter Rolle angezeigt.

## Um die Rollen von Benutzern in einem Projekt zu ändern

1. Navigieren Sie zu dem Projekt, in dem Sie die den Projektmitgliedern zugewiesenen Rollen ändern möchten.

### Tip

In der oberen Navigationsleiste können Sie auswählen, welches Projekt angezeigt werden soll.

2. Wählen Sie im Navigationsbereich die Option Projekteinstellungen aus.
3. Wählen Sie auf der Registerkarte Mitglieder unter Projektmitglieder die Benutzer aus, deren Rollen Sie ändern möchten. Wählen Sie Aktion und dann Rolle bearbeiten aus.
4. Wählen Sie unter Rolle die Projektrolle aus und klicken Sie dann auf Bestätigen.

## Rollen im Space anzeigen und ändern

Alle Benutzer, die Einladungen zu einem Projekt annehmen, CodeCatalyst werden Mitglieder des Projektbereichs. Sie können die Liste der Space-Mitglieder einsehen. Sie können die Rollen der Benutzer von Eingeschränkter Zugriff auf Space-Administrator ändern, um Ihren Bereich und die zugehörigen Ressourcen besser verwalten zu können. Die Rolle des Space-Administrators ist die einzige Rolle, in der Benutzer Projekte erstellen können CodeCatalyst.

### Warning

Die Rolle des Space-Administrators ist die mächtigste Rolle in CodeCatalyst. Benutzer mit dieser Rolle können jede beliebige Aktion ausführen CodeCatalyst, einschließlich des Löschens des Bereichs. Weisen Sie diese Rolle nur Benutzern zu, die diese Zugriffsebene für Ihren Bereich benötigen. Weitere Informationen finden Sie unter [Rolle „Space-Administrator“](#).

## Um die Rolle eines Benutzers in dem Bereich zu ändern

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu dem Bereich.

**Tip**

Wenn Sie zu mehr als einem Bereich gehören, können Sie in der oberen Navigationsleiste auswählen, welcher Bereich angezeigt werden soll.

3. Wählen Sie den Tab Mitglieder.
4. Wählen Sie den Benutzer aus, dessen Rolle Sie ändern möchten, und klicken Sie dann auf Rolle ändern.
5. Wählen Sie unter Rolle ändern die Rolle aus, die Sie zuweisen möchten, und klicken Sie dann auf Bestätigen.

## Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst

Um auf einige CodeCatalyst Ressourcen, z. B. Quell-Repositorys, auf einem lokalen Computer mit einem Git-Client oder einer integrierten Entwicklungsumgebung (IDE) zuzugreifen, müssen Sie ein anwendungsspezifisches Passwort eingeben. Sie können ein persönliches Zugriffstoken (PAT) erstellen, das Sie für diesen Zweck verwenden können. Von Ihnen erstellte PATs werden in allen Bereichen und Projekten in CodeCatalyst Ihrer Benutzeridentität verknüpft. Sie können mehr als eine PAT für Ihre CodeCatalyst Identität erstellen.

Sie können die Namen und das Ablaufdatum der PATs einsehen, die Sie erstellt haben, und Sie können die PATs löschen, die Sie nicht mehr benötigen. Sie können das PAT-Geheimnis nur zu dem Zeitpunkt kopieren, zu dem Sie es erstellen.

**Note**

Standardmäßig laufen PATs in einem Jahr ab.

### PATs erstellen

PATs sind mit Ihrer Benutzeridentität in verknüpft. CodeCatalyst Sie können ein PAT-Geheimnis nur zu dem Zeitpunkt kopieren, zu dem Sie es erstellen.

#### PATs erstellen (Konsole)

Sie können die Konsole verwenden, um PATs in zu erstellen. CodeCatalyst

## Um ein persönliches Zugriffstoken (Konsole) zu erstellen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie in der oberen Menüleiste Ihr Profil-Badge und dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.

### Tip

Sie können Ihr Benutzerprofil auch finden, indem Sie die Mitgliederseite für ein Projekt oder einen Bereich aufrufen und Ihren Namen aus der Mitgliederliste auswählen.

3. Wählen Sie unter Persönliche Zugriffstoken die Option Erstellen aus.

Die Seite „PAT erstellen“ wird angezeigt.

4. Geben Sie im Feld PAT-Name einen aussagekräftigen Namen für Ihre PAT ein.
5. Behalten Sie im Feld Ablaufdatum das Standarddatum bei oder wählen Sie das Kalendersymbol, um ein benutzerdefiniertes Datum auszuwählen. Das Ablaufdatum liegt standardmäßig auf einem Jahr ab dem aktuellen Datum.
6. Wählen Sie Erstellen.

### Tip

Sie können dieses Token auch erstellen, wenn Sie Repository klonen für ein Quell-Repository wählen.

7. Um das PAT-Geheimnis zu kopieren, wählen Sie Kopieren. Speichern Sie das PAT-Geheimnis dort, wo Sie es abrufen können.

### Important

Das PAT-Geheimnis wird nur einmal angezeigt. Sie können es nicht abrufen, nachdem Sie das Fenster geschlossen haben. Wenn Sie das PAT-Geheimnis nicht an einem sicheren Ort gespeichert haben, können Sie ein anderes erstellen.

## PATs erstellen (CLI)

Sie können die CLI verwenden, um PATs in CodeCatalyst zu erstellen.

Um ein persönliches Zugriffstoken zu erstellen ( )AWS CLI

1. Führen Sie den Befehl im Terminal oder in der create-access-token Befehlszeile wie folgt aus.

```
aws codecatalyst create-access-token
```

Bei Erfolg gibt der Befehl wie im folgenden Beispiel Informationen über die erstellte PAT zurück.

```
{
  "secret": "value",
  "name": "marymajor-22222EXAMPLE",
  "expiresTime": "2024-02-04T01:56:04.402000+00:00"
}
```

- 2.

Sie können das PAT-Geheimnis nur einmal anzeigen, nämlich wenn Sie das PAT erstellen. Wenn Sie ein PAT-Geheimnis verloren haben oder befürchten, dass es nicht sicher gespeichert ist, können Sie ein neues erstellen.

Sie können die mit Ihrem Benutzerkonto verknüpften PATs einsehen, indem Sie den verwenden. AWS CLI Sie können nur Informationen über das PAT einsehen, nicht aber den Wert des PAT-Geheimnisses selbst.

### Note

Stellen Sie sicher, dass Sie eine aktuelle Version von verwenden, mit der AWS CLI Sie arbeiten können CodeCatalyst. Frühere Versionen enthalten die CodeCatalyst Befehle möglicherweise nicht. Sie müssen Ihr AWS CLI Profil konfigurieren, bevor Sie es mit verwenden können CodeCatalyst. Weitere Informationen finden Sie unter [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#).

## PATs anzeigen

Sie können PATs in anzeigen. CodeCatalyst In der Liste werden alle PATs angezeigt, die Sie mit Ihrer Benutzeridentität verknüpft haben. Ihre PAT ist in allen Bereichen und Projekten in CodeCatalyst Ihrem Benutzerprofil verknüpft. Abgelaufene PATs werden nicht angezeigt, da sie nach ihrem Ablauf gelöscht werden.

### PATs anzeigen (Konsole)

Sie können die Konsole verwenden, um PATs anzuzeigen, die mit Ihrer Benutzeridentität verknüpft sind, in. CodeCatalyst

Um Ihre persönlichen Zugriffstoken einzusehen (Konsole)

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie in der oberen Menüleiste Ihr Profil-Badge und dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.

#### Tip

Sie können Ihr Benutzerprofil auch finden, indem Sie die Mitgliederseite für ein Projekt oder einen Bereich aufrufen und Ihren Namen aus der Mitgliederliste auswählen.

3. Unter Persönliche Zugriffstoken findest du die Namen und Ablaufdaten deiner aktuellen PATs.

### PATs anzeigen (CLI)

Sie können die CLI verwenden, um PATs anzuzeigen, die Ihrer Benutzeridentität in CodeCatalyst zugeordnet sind.

Um Ihre persönlichen Zugriffstoken einzusehen ()AWS CLI

- Führen Sie den Befehl im Terminal oder in der list-access-tokens Befehlszeile wie folgt aus.

```
aws codecatalyst list-access-tokens
```

Bei Erfolg gibt der Befehl wie im folgenden Beispiel Informationen über die mit Ihrem Benutzerkonto verknüpften PATs zurück.

```
{
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaa",
      "name": "marymajor-22222EXAMPLE",
      "expiresTime": "2024-02-04T01:56:04.402000+00:00"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEbbbb",
      "name": "marymajor-11111EXAMPLE",
      "expiresTime": "2023-03-12T01:58:40.694000+00:00"
    }
  ]
}
```

## PATs löschen

Sie können PATs löschen, die mit Ihrer Benutzeridentität verknüpft sind, in CodeCatalyst

### PATs löschen (Konsole)

Sie können die Konsole verwenden, um PATs in zu löschen. CodeCatalyst

Um ein persönliches Zugriffstoken (Konsole) zu löschen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie in der oberen Menüleiste Ihr Profil-Badge und dann Meine Einstellungen aus. Die Seite CodeCatalyst Meine Einstellungen wird geöffnet.

#### Tip

Sie können Ihr Benutzerprofil auch finden, indem Sie die Mitgliederseite für ein Projekt oder einen Bereich aufrufen und Ihren Namen aus der Mitgliederliste auswählen.

3. Wählen Sie unter Persönliche Zugriffstoken den Selektor neben dem PAT aus, den Sie löschen möchten, und wählen Sie dann Löschen aus.

<name>Auf dem Feld „PAT löschen“: Geben Sie auf der Seite delete in das Textfeld ein, um den Löschvorgang zu bestätigen. Wählen Sie Löschen aus.



## PATs löschen (CLI)

Sie können eine mit Ihrer Benutzeridentität verknüpfte PAT löschen, indem Sie die AWS CLI verwenden. Dazu müssen Sie die ID für die PAT angeben, die Sie mit dem `delete-access-token` Befehl anzeigen können.

### Note

Stellen Sie sicher, dass Sie eine aktuelle Version von verwenden, mit der AWS CLI Sie arbeiten können CodeCatalyst. Frühere Versionen enthalten die CodeCatalyst Befehle möglicherweise nicht. Weitere Hinweise zur Verwendung von AWS CLI with CodeCatalyst finden Sie unter [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#).

So löschen Sie ein persönliches Zugriffstoken (AWS CLI)

- Führen Sie den Befehl im Terminal oder in der `delete-access-token` Befehlszeile aus und geben Sie die ID für die PAT an, die Sie löschen möchten. Führen Sie beispielsweise den folgenden Befehl aus, um eine PAT mit der ID `123EXAMPLE` zu löschen.

```
aws codecatalyst delete-access-token --id a1b2c3d4-5678-90ab-cdef-EXAMPLEebbbb
```

Bei Erfolg gibt dieser Befehl keine Antwort zurück.

## Multi-Faktor-Authentifizierung (MFA) bei Amazon CodeCatalyst

Unabhängig davon, ob Sie Ihr AWS Builder ID-Profil für den privaten oder beruflichen Gebrauch erstellt haben, empfehlen wir Ihnen, die Multi-Faktor-Authentifizierung (MFA) als weitere Sicherheitsebene zu konfigurieren. Wir empfehlen insbesondere, MFA zu konfigurieren, wenn Sie Mitglied eines Bereichs sind und mit anderen an Projekten zusammenarbeiten. Da mehr als eine Person Zugriff auf ein Projekt haben kann, gibt es mehr Möglichkeiten für Sicherheitsverletzungen.

Wenn Sie MFA aktivieren, müssen Sie sich CodeCatalyst mit Ihrer E-Mail-Adresse und Ihrem Passwort bei Amazon anmelden. Dieser Teil der Anmeldung ist der erste Faktor, wenn Sie etwas verwenden, das Sie kennen. Anschließend melden Sie sich entweder mit einem Code oder einem Sicherheitsschlüssel an. Dies ist der zweite Faktor, den Sie haben. Der zweite Faktor könnte ein Authentifizierungscode sein, der entweder von Ihrem Mobilgerät oder durch Tippen oder Drücken

eines mit Ihrem Computer verbundenen Sicherheitsschlüssels generiert wird. Zusammengenommen sorgen diese verschiedenen Faktoren für mehr Sicherheit, indem sie unbefugten Zugriff verhindern.

## Wie registriere ich ein Gerät für die Verwendung mit Multi-Faktor-Authentifizierung

Gehen Sie wie folgt vor: Mein Profil > Multi-Faktor-Authentifizierung, um Ihr neues Gerät für die Multi-Faktor-Authentifizierung (MFA) zu registrieren.

### Note

Wir empfehlen, dass Sie zuerst die entsprechende Authentifizierungs-App auf Ihr Gerät herunterladen, bevor Sie mit den Schritten in diesem Verfahren beginnen. Eine Liste der Apps, die Sie für MFA-Geräte verwenden können, finden Sie unter [Authentifizierungsanwendungen](#).

Um Ihr Gerät für die Verwendung mit MFA zu registrieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Klicken Sie rechts oben auf den Pfeil neben dem Symbol mit Ihrer ersten Initiale und wählen Sie dann Benutzerprofil aus. Die CodeCatalyst Profilseite wird geöffnet.
3. Wählen Sie auf der Profilseite die Option Profil und Sicherheit verwalten aus. Die AWS Builder ID-Profilseite wird geöffnet.
4. Wählen Sie links auf der Seite Sicherheit aus.
5. Wählen Sie auf der Seite Multi-Faktor-Authentifizierung die Option Gerät registrieren aus.
6. Wählen Sie auf der Seite MFA-Gerät registrieren einen der folgenden MFA-Gerätetypen aus und folgen Sie den Anweisungen:
  - Sicherheitsschlüssel oder integrierter Authentifikator
    1. Folgen Sie auf der Seite Sicherheitsschlüssel Ihres Benutzers registrieren den Anweisungen Ihres Browsers oder Ihrer Plattform.

### Note

Diese Erfahrung hängt von Ihrem Betriebssystem und Browser ab. Folgen Sie daher den Anweisungen, die von Ihrem Browser oder Ihrer Plattform angezeigt werden.

Nachdem Ihr Gerät erfolgreich registriert wurde, haben Sie die Möglichkeit, Ihrem neu registrierten Gerät einen benutzerfreundlichen Anzeigenamen zuzuweisen. Wenn Sie dies ändern möchten, wählen Sie Umbenennen, geben Sie den neuen Namen ein und wählen Sie dann Speichern.

- Authenticator-App

1. Auf der Seite Authentifikator-App einrichten finden Sie möglicherweise Konfigurationsinformationen für das neue MFA-Gerät, einschließlich einer QR-Code-Grafik. Die Grafik ist eine Darstellung des geheimen Schlüssels, der für die manuelle Eingabe auf Geräten verfügbar ist, die QR-Codes nicht unterstützen.
2. Gehen Sie mit dem physischen MFA-Gerät wie folgt vor:
  - a. Öffnen Sie eine kompatible MFA-Authenticator-App. Eine Liste der getesteten Apps, die Sie mit MFA-Geräten verwenden können, finden Sie unter [Getestete Authenticator-Apps](#). Wenn die MFA-App mehrere Geräte unterstützt, wählen Sie die Option zum Erstellen eines neuen MFA-Geräts.
  - b. Stellen Sie fest, ob die MFA-App QR-Codes unterstützt, und führen Sie dann auf der Seite Authenticator-App einrichten einen der folgenden Schritte aus:
    - i. Wählen Sie Show QR code (QR-Code anzeigen) und verwenden Sie anschließend die App, um den QR-Code zu scannen. Sie können beispielsweise das Kamerasymbol oder eine ähnliche Option wie Scan code (Code scannen) auswählen. Verwenden Sie anschließend die Kamera des Geräts, um den Code zu scannen.
    - ii. Wählen Sie Geheimen Schlüssel anzeigen und geben Sie dann diesen geheimen Schlüssel in Ihre MFA-App ein.

 **Important**

Wenn Sie ein MFA-Gerät für AWS Builder ID konfigurieren, speichern Sie eine Kopie des QR-Codes oder des geheimen Schlüssels an einem sicheren Ort. Dies kann helfen, wenn Sie das Telefon verlieren oder die MFA-Authenticator-App neu installieren müssen. Wenn eines dieser Dinge eintritt, können Sie die App schnell neu konfigurieren, um dieselbe MFA-Konfiguration zu verwenden.

3. Geben Sie auf der Seite Authenticator-App einrichten unter Authenticator-Code das Einmalpasswort ein, das derzeit auf dem physischen MFA-Gerät angezeigt wird.

**⚠ Important**

Senden Sie die Anforderung direkt nach der Erzeugung der Codes. Wenn Sie den Code generieren und dann zu lange warten, um die Anfrage einzureichen, wurde das MFA-Gerät erfolgreich mit Ihrem AWS Builder ID-Profil verknüpft, aber das MFA-Gerät ist nicht synchronisiert. Dies liegt daran, weil die zeitgesteuerten Einmalpasswörter (TOTP) nach einer kurzen Zeit ungültig werden. In diesem Fall können Sie das Gerät neu synchronisieren.

4. Klicken Sie auf Assign MFA (MFA zuordnen). Das MFA-Gerät kann jetzt mit der Generierung von Einmalpasswörtern beginnen und ist jetzt einsatzbereit.

## Authentifizierungsanwendungen

Bei Authenticator-Apps handelt es sich um Authentifikatoren von Drittanbietern, die auf Einmalkennwörtern (OTP) basieren. Benutzer können eine auf ihrem Mobilgerät oder Tablet installierte Authentifizierungsanwendung als autorisiertes MFA-Gerät verwenden. Die Authentifizierungsanwendung eines Drittanbieters muss mit RFC 6238 konform sein, einem standardbasierten TOTP-Algorithmus (Time-based One-Time Password), der sechsstellige Authentifizierungs-codes generieren kann.

Wenn Benutzer zur Eingabe von MFA aufgefordert werden, müssen sie einen gültigen Code aus ihrer Authenticator-App in das angezeigte Eingabefeld eingeben. Jedes MFA-Gerät, das einem Benutzer zugeordnet ist, muss eindeutig sein. Für jeden Benutzer können zwei Authentifizierungs-Apps registriert werden.

### Getestete Authenticator-Apps

Obwohl jede TOTP-konforme Anwendung mit IAM Identity Center MFA funktioniert, sind in der folgenden Tabelle bekannte Authentifikator-Apps von Drittanbietern aufgeführt, aus denen Sie wählen können.

| Betriebssystem | Getestete Authentifizierungs-App   |
|----------------|--|
| Android        | <a href="#">Authy</a> , <a href="#">Duo Mobile</a> , <a href="#">Authentifikator</a> , <a href="#">Microsoft LastPass Authenticator</a> , <a href="#">Google Authenticator</a> |

|                |  |
|----------------|--|
| Betriebssystem | Getestete Authentifizierungs-App   |
| iOS            | <a href="#">Authy</a> , <a href="#">Duo Mobile</a> , <a href="#">Authentifikator</a> , <a href="#">Microsoft LastPass Authenticator</a> , <a href="#">Google Authenticator</a> |

## Ihre MFA-Geräte ändern

Nachdem Sie ein MFA-Gerät registriert haben, können Sie seinen Namen ändern oder es löschen. Wir empfehlen, immer mindestens ein MFA-Gerät zu aktivieren, um eine zusätzliche Sicherheitsebene zu gewährleisten. Sie können bis zu fünf Geräte registrieren lassen. Informationen darüber, wie Sie weitere hinzufügen können, finden Sie unter [Wie registriere ich ein Gerät für die Verwendung mit Multi-Faktor-Authentifizierung](#).

### Ein MFA-Gerät umbenennen

So benennen Sie Ihr MFA-Gerät um

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Klicken Sie rechts oben auf den Pfeil neben dem Symbol mit Ihrer ersten Initialen und wählen Sie dann Benutzerprofil aus. Die CodeCatalyst Profilsseite wird geöffnet.
3. Wählen Sie auf der Profilsseite die Option Profil und Sicherheit verwalten aus. Die AWS Builder ID-Profilesseite wird geöffnet.
4. Wählen Sie links auf der Seite die Option Multi-Faktor-Authentifizierung aus. Sie werden sehen, dass Umbenennen ausgegraut ist, wenn Sie auf der Seite ankommen.
5. Wählen Sie das MFA-Gerät aus, das Sie ändern möchten. Wählen Sie Rename (Umbenennen) aus. Dann erscheint ein Modal.
6. Geben Sie in der sich öffnenden Aufforderung den neuen Namen in das Feld MFA-Gerätename ein und wählen Sie dann Umbenennen. Das umbenannte Gerät wird unter Multi-Factor Authentication Devices (MFA) angezeigt.

### Löschen eines MFA-Geräts

So löschen Sie ein MFA-Gerät

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

2. Klicken Sie rechts oben auf den Pfeil neben dem Symbol mit Ihrer ersten Initiale und wählen Sie dann Benutzerprofil aus. Die CodeCatalyst Profilseite wird geöffnet.
3. Wählen Sie auf der Profilseite die Option Profil und Sicherheit verwalten aus. Die AWS Builder ID-Profilseite wird geöffnet.
4. Wählen Sie links auf der Seite die Option Multi-Faktor-Authentifizierung aus. Sie werden sehen, dass Löschen ausgegraut ist, wenn Sie auf der Seite ankommen.
5. Wählen Sie das MFA-Gerät aus, das Sie ändern möchten. Wählen Sie Löschen aus. Es erscheint ein Modal mit der Aufschrift MFA-Gerät löschen? . Folgen Sie den Anweisungen, um Ihr Gerät zu löschen.
6. Wählen Sie Löschen aus. Das gelöschte Gerät wird nicht mehr unter Multi-Factor Authentication Devices (MFA) angezeigt.

## Sicherheit bei Amazon CodeCatalyst

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt ist, die Anforderungen der sicherheitsempfindlichsten Bereiche zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der übergreifenden Verantwortlichkeit](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) . Weitere Informationen zu den Compliance-Programmen, die für gelten CodeCatalyst, finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#) .
- Sicherheit in der Cloud — Ihre Verantwortung hängt von den AWS Diensten ab, die Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von Amazon anwenden können CodeCatalyst. Es zeigt Ihnen, wie Sie die Konfiguration vornehmen CodeCatalyst , um Ihre Sicherheits- und Compliance-Ziele zu erreichen.

Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer CodeCatalyst Ressourcen unterstützen.

## Inhalt

- [Datenschutz bei Amazon CodeCatalyst](#)
- [Identity and Access Management und Amazon CodeCatalyst](#)
- [Konformitätsvalidierung für Amazon CodeCatalyst](#)
- [Resilienz bei Amazon CodeCatalyst](#)
- [Infrastruktursicherheit bei Amazon CodeCatalyst](#)
- [Konfiguration und Schwachstellenanalyse in Amazon CodeCatalyst](#)
- [Ihre Daten und Ihr Datenschutz bei Amazon CodeCatalyst](#)
- [Bewährte Methoden für Workflow-Aktionen in Amazon CodeCatalyst](#)
- [CodeCatalyst Vertrauensmodell](#)

## Datenschutz bei Amazon CodeCatalyst

CodeCatalyst ist, wie in diesem Modell beschrieben, für den Schutz der globalen Infrastruktur für den Service verantwortlich. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieses Modell der gemeinsamen Verantwortung gilt für den Datenschutz in CodeCatalyst.

Aus Datenschutzgründen empfehlen wir Ihnen, Ihre Kontoanmeldeinformationen zu schützen und bei der Anmeldung eine Multi-Faktor-Authentifizierung einzurichten. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung \(MFA\) bei Amazon CodeCatalyst](#).

Geben Sie keine vertraulichen oder vertraulichen Informationen, wie z. B. die E-Mail-Adressen Ihrer Kunden, in Stichwörter oder Freiformfelder wie ein Namensfeld ein. Dazu gehören Ressourcennamen und alle anderen Kennungen, die Sie eingeben, zusätzlich zu allen verbundenen Kennungen. CodeCatalyst AWS-Konten Geben Sie beispielsweise keine vertraulichen oder sensiblen Informationen als Teil von Raum-, Projekt- oder Einsatzflottennamen ein. Alle Daten, die Sie in Tags, Namen oder Freiformfelder für Namen eingeben, können für Abrechnungs- oder Diagnoseprotokolle verwendet oder in URL-Pfaden enthalten sein. Dies gilt für die Verwendung der Konsole, der APIAWS CLI, des CodeCatalyst Action Development Kit oder anderer AWS SDKs.

Wenn Sie eine URL für einen externen Server angeben, empfehlen wir dringend, dass Sie in der URL keine Informationen zu Sicherheitsanmeldedaten angeben, um Ihre Anfrage an diesen Server zu überprüfen.

CodeCatalyst Quell-Repositorys werden im Ruhezustand automatisch verschlüsselt. Es ist kein Eingreifen des Kunden erforderlich. CodeCatalyst verschlüsselt auch Repository-Daten während der Übertragung mithilfe des HTTPS-Protokolls.

CodeCatalyst unterstützt MFA. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung \(MFA\) bei Amazon CodeCatalyst](#).

## Datenverschlüsselung

CodeCatalyst speichert und überträgt Daten sicher innerhalb des Dienstes. Alle Daten werden bei der Übertragung und im Ruhezustand verschlüsselt. Alle vom Dienst erstellten oder gespeicherten Daten, einschließlich aller Metadaten für den Dienst, werden nativ im Dienst gespeichert und verschlüsselt.

### Note

Informationen zu Problemen werden zwar sicher innerhalb des Dienstes gespeichert, Informationen zu offenen Problemen werden jedoch auch im lokalen Cache des Browsers gespeichert, in dem Sie Problemforen, Backlogs und einzelne Probleme angesehen haben. Um optimale Sicherheit zu gewährleisten, sollten Sie Ihren Browser-Cache leeren, um diese Informationen zu entfernen.

Wenn Sie Ressourcen verwenden, die mit verknüpft sind CodeCatalyst, z. B. eine Kontoverbindung zu einem AWS-Konto oder einem verknüpften Repository in GitHub, werden Daten, die von CodeCatalyst dieser verknüpften Ressource übertragen werden, verschlüsselt, aber die Datenverarbeitung in dieser verknüpften Ressource wird von diesem verknüpften Dienst verwaltet. Weitere Informationen finden Sie in der Dokumentation zum verknüpften Dienst und [Bewährte Methoden für Workflow-Aktionen in Amazon CodeCatalyst](#).

## Schlüsselverwaltung

CodeCatalyst unterstützt keine Schlüsselverwaltung.



## Datenschutz für den Datenverkehr zwischen Netzwerken

Wenn Sie einen Bereich in erstellen CodeCatalyst, wählen Sie aus, AWS-Region wo die Daten und Ressourcen für diesen Bereich gespeichert werden sollen. Projektdaten und Metadaten verlassen diesen Bereich nie AWS-Region. Um jedoch die Navigation innerhalb der Partition zu unterstützen CodeCatalyst, wird ein begrenzter Satz von Speicherplatz-, Projekt- und Benutzermetadaten auf alle AWS-Regionen [Partitionen](#) repliziert. Es wird nicht AWS-Regionen außerhalb dieser Partition repliziert. Wenn Sie beispielsweise AWS-Region bei der Erstellung Ihres Bereichs die Option USA West (Oregon) auswählen, werden Ihre Daten nicht in Regionen in China repliziert oder. AWS GovCloud (US) Weitere Informationen finden Sie unter [Verwaltung AWS-Regionen](#), [AWSglobale Infrastruktur](#) und [AWSService-Endpunkte](#).

Zu den Daten, die AWS-Regionen innerhalb einer Partition repliziert werden, gehören:

- Ein verschlüsselter Hashwert, der den Namen des Bereichs darstellt, um die Eindeutigkeit der Bereichsnamen sicherzustellen. Dieser Wert ist für Menschen nicht lesbar und gibt nicht die tatsächlichen Namen von Leerzeichen preis
- Die eindeutige ID des Bereichs
- Metadaten für den Bereich, die bei der Navigation zwischen Räumen helfen
- Der AWS-Region Ort, an dem sich der Bereich befindet
- Die eindeutigen IDs aller Projekte im Space
- Die Rollen-ID, die die Rolle eines Benutzers in einem Bereich oder Projekt angibt
- Bei der Registrierung für CodeCatalyst Daten und Metadaten zum Anmeldevorgang, einschließlich:
  - Die eindeutige ID des AWS Builder ID
  - Der Anzeigename für den Benutzer in ihrem AWS Builder ID
  - Der Alias des Benutzers in ihrem AWS Builder ID
  - Die E-Mail-Adresse, die verwendet wurde, als sich der Benutzer für sein Konto angemeldet hat AWS Builder ID
  - Der Fortschritt des Anmeldevorgangs
  - Wenn Sie im Rahmen des Anmeldevorgangs einen Bereich erstellen, die AWS-Konto ID, die als Rechnungskonto für den Bereich verwendet wird

Raumnamen sind überall einzigartig CodeCatalyst. Achten Sie darauf, dass der Name des Bereichs keine sensiblen Daten enthält.

Wenn Sie mit verknüpften Ressourcen und verbundenen Konten arbeiten, z. B. mit einer Verbindung zu einem AWS-Konto oder einem GitHub Repository, empfehlen wir, Ihre Quell- und Zielspeicherorte mit der höchsten Sicherheitsstufe zu konfigurieren, die von jedem Standort unterstützt wird.

CodeCatalyst sichert die Verbindung zwischen AWS-Konten AWS-Regionen, und Availability Zones mithilfe von Transport Layer Security (TLS) 1.2.

## Identity and Access Management und Amazon CodeCatalyst

In Amazon erstellen und verwenden Sie eine AWS -Entwickler-ID CodeCatalyst, um sich anzumelden und auf Ihre Bereiche und Projekte zuzugreifen. Eine AWS -Entwickler-ID ist keine Identität in AWS Identity and Access Management (IAM) und existiert nicht in einem AWS-Konto. CodeCatalyst integriert sich jedoch in IAM, wenn ein Speicherplatz für Fakturierungszwecke verifiziert wird und wenn eine Verbindung zu einem hergestellt wird, AWS-Konto um Ressourcen in diesem zu erstellen und zu verwenden AWS-Konto.

AWS Identity and Access Management (IAM) ist ein AWS-Service, mit dem Administratoren den Zugriff auf AWS-Ressourcen sicher steuern können. IAM-Administratoren steuern, wer für die Nutzung von Ressourcen authentifiziert (angemeldet) und autorisiert (über Berechtigungen verfügen) werden kann. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Wenn Sie einen Bereich in Amazon erstellen CodeCatalyst, müssen Sie einen AWS-Konto als Fakturierungskonto für Ihren Bereich verbinden. Sie müssen über Administratorberechtigungen in der verfügen, AWS-Konto um den CodeCatalyst Speicherplatz zu überprüfen, oder über die -Berechtigung. Sie haben auch die Möglichkeit, eine IAM-Rolle für Ihren Bereich hinzuzufügen, die verwenden CodeCatalyst kann, um Ressourcen in diesem verbundenen zu erstellen und darauf zuzugreifen AWS-Konto. Dies wird als [Servicerolle](#) bezeichnet. Sie können Verbindungen zu mehr als einem AWS-Konto und Servicerollen für CodeCatalyst in jedem dieser Konten erstellen.

### Note

Die Fakturierung für CodeCatalyst erfolgt in der , die als Fakturierungskonto AWS-Konto festgelegt ist. Wenn Sie jedoch eine CodeCatalyst Servicerolle in diesem AWS-Konto oder in einem anderen verbundenen erstellen AWS-Konto, werden Ressourcen, die von der CodeCatalyst Servicerolle erstellt und verwendet werden, in diesem verbundenen abgerechnet AWS-Konto. Weitere Informationen finden Sie unter [Verwalten der Fakturierung](#) im Amazon- CodeCatalyst Administratorhandbuch.

## Themen

- [Identitätsbasierte Richtlinien in IAM](#)
- [Richtlinienaktionen in IAM](#)
- [Richtlinienressourcen in IAM](#)
- [Richtlinienbedingungsschlüssel in IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für CodeCatalyst Verbindungen](#)
- [Verwenden von Tags zur Steuerung des Zugriffs auf Kontoverbindungsressourcen](#)
- [CodeCatalyst -Berechtigungsreferenz](#)
- [Verwenden von serviceverknüpften Rollen für CodeCatalyst](#)
- [AWS Von verwaltete Richtlinien für Amazon CodeCatalyst](#)
- [IAM-Rollen für Amazon- CodeCatalyst Zugriff auf -AWSRessourcen](#)

## Identitätsbasierte Richtlinien in IAM

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können. Bei dieser Identität kann es sich um einen Benutzer, eine Benutzergruppe oder eine Rolle handeln. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für CodeCatalyst

Beispiele für CodeCatalyst identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für CodeCatalyst Verbindungen](#).

## Richtlinienaktionen in IAM

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal kann welche Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keine passende API-Operation gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "prefix:action1",  
  "prefix:action2"  
]
```

## Richtlinienressourcen in IAM

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal kann welche Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*" 
```

## Richtlinienbedingungsschlüssel in IAM

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal kann welche Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet AWS die Bedingung mittels einer logischen OR-Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Eine Liste aller globalen AWS-Bedingungsschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

## Beispiele für identitätsbasierte Richtlinien für CodeCatalyst Verbindungen

In AWS-Konten sind erforderlich CodeCatalyst, um die Abrechnung für einen Bereich zu verwalten und auf Ressourcen in Projektworkflows zuzugreifen. Eine Kontoverbindung wird verwendet, um das Hinzufügen zu einem Bereich AWS-Konten zu autorisieren. Identitätsbasierte Richtlinien werden in der verbundenen verwendet AWS-Konten.

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, CodeCatalyst-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator muss diese Richtlinien anschließend den Benutzern anfügen, die sie benötigen.

Die folgenden IAM-Beispielrichtlinien gewähren Berechtigungen für Aktionen im Zusammenhang mit Kontoverbindungen. Verwenden Sie sie, um den Zugriff für die Verbindung von Konten mit zu beschränken CodeCatalyst.

Beispiel 1: Einem Benutzer erlauben, Verbindungsanforderungen in einem einzigen zu akzeptieren AWS-Region

Die folgende Berechtigungsrichtlinie erlaubt es Benutzern nur, Anforderungen für Verbindungen zwischen und anzuzeigen CodeCatalyst und zu akzeptierenAWS-Konten. Darüber hinaus verwendet die Richtlinie eine -Bedingung, um nur die Aktionen in der Region us-west-2 und nicht von anderen zuzulassenAWS-Regionen. Um die Anforderung anzuzeigen und zu genehmigen, meldet sich der Benutzer bei der AWS Management Console mit demselben Konto an, das in der Anforderung angegeben ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:GetPendingConnection"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    }
  ]
}
```

Beispiel 2: Zulassen der Verwaltung von Verbindungen in der Konsole für ein einzelnes AWS-Region

Mit der folgenden Berechtigungsrichtlinie können Benutzer Verbindungen zwischen CodeCatalyst und AWS-Konten in einer einzelnen Region verwalten. Die Richtlinie verwendet eine Bedingung, um nur die Aktionen in der Region us-west-2 und nicht von anderen zuzulassenAWS-Regionen. Nachdem Sie eine Verbindung erstellt haben, können Sie die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle erstellen, indem Sie die Option in der auswählenAWS Management Console. In der Beispielrichtlinie enthält die Bedingung für die

iam:PassRole Aktion die Service-Prinzipale für CodeCatalyst. Nur Rollen mit diesem Zugriff werden in der erstellten AWS Management Console.

```
{
  "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"
          ]
        }
      }
    }
  ]
}
```

```
]
}
```

### Beispiel 3: Verweigern der Verwaltung von Verbindungen

Die folgende Berechtigungsrichtlinie verweigert Benutzern die Möglichkeit, Verbindungen zwischen CodeCatalyst und zu verwaltenAWS-Konten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*"
    }
  ]
}
```

### Verwenden von Tags zur Steuerung des Zugriffs auf Kontoverbindungsressourcen

Tags können an die Ressource angehängt oder in der Anfrage an Dienste weitergegeben werden, die Tagging unterstützen. Ressourcen in Richtlinien können Tags enthalten, und einige Aktionen in Richtlinien können Tags enthalten. Zu den Bedingungsschlüsseln für Tagging gehören die `aws:ResourceTag` Bedingungsschlüssel `aws:RequestTag` und. Wenn Sie eine IAM-Richtlinie erstellen, können Sie Markierungs-Bedingungsschlüssel verwenden, um Folgendes zu kontrollieren:

- Welche Benutzer auf der Grundlage von Tags, über die sie bereits verfügt, Aktionen an einer Verbindungsressource ausführen können.
- Welche Tags in der Anforderung einer Aktion übergeben werden können.
- Ob bestimmte Tag-Schlüssel in einer Anforderung verwendet werden können.

Die folgenden Beispiele zeigen, wie Tag-Bedingungen in Richtlinien für CodeCatalyst Kontoverbindungsbenutzer festgelegt werden. Weitere Informationen über Bedingungsschlüssel finden Sie unter [Richtlinienbedingungsschlüssel in IAM](#).



## Beispiel 1: Erlauben Sie Aktionen, die auf Tags in der Anfrage basieren

Die folgende Richtlinie gewährt Benutzern die Erlaubnis, Kontoverbindungen zu genehmigen.

Hierfür werden die Aktionen `AcceptConnection` und `TagResource` zugelassen, wenn die Anforderung ein Tag mit dem Namen `Project` und dem Wert `ProjectA` angibt. (Der Bedingungsschlüssel `aws:RequestTag` wird verwendet, um zu steuern, welche Tags in einer IAM-Anforderung übergeben werden können.) Die Bedingung `aws:TagKeys` stellt sicher, dass bei Tag-Schlüsseln die Groß- und Kleinschreibung beachtet wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

## Beispiel 2: Aktionen auf der Grundlage von Ressourcen-Tags zulassen

Die folgende Richtlinie gewährt Benutzern die Erlaubnis, Aktionen an Kontoverbindungsressourcen durchzuführen und Informationen darüber abzurufen.

Zu diesem Zweck sind bestimmte Aktionen zulässig, wenn die Verbindung über ein Tag verfügt, das `Project` mit dem Wert `benannt istProjectA`. (Der Bedingungsschlüssel `aws:ResourceTag` wird verwendet, um zu steuern, welche Tags in einer IAM-Anforderung übergeben werden können.)

```
{
```

```

"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 -Berechtigungsreferenz

Dieser Abschnitt enthält eine Berechtigungsreferenz für Aktionen, die mit der Kontoverbindungsressource für verwendet werden AWS-Konten, die mit verbunden sind CodeCatalyst. Im folgenden Abschnitt werden reine Berechtigungsaktionen beschrieben, die sich auf die Verbindung von Konten beziehen.

### Erforderliche Berechtigungen für Kontoverbindungen

Die folgenden Berechtigungen sind für die Arbeit mit Kontoverbindungen erforderlich.

| CodeCatalyst -Berechtigungen für Kontoverbindungen | Erforderliche Berechtigungen   | Ressourcen  |
|--|--|---|
| AcceptConnection                                   | Erforderlich, um eine Anforderung zum Verbinden dieses Kontos mit einem CodeCatalyst Bereich zu akzeptieren. Dies ist nur eine | Unterstützt nur einen Platzhalter (*) im Resource-Richtlinienelement. |

| CodeCatalyst -Berechtigungen für Kontoverbindungen | Erforderliche Berechtigungen   | Ressourcen   |
|--|--|--|
|  | IAM-Richtlinienberechtigung, keine API-Aktion.   |  |
| AssociateIamRoleToConnection                       | Erforderlich, um eine IAM-Rolle einer Kontoverbindung zuzuordnen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                     | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| DeleteConnection                                   | Erforderlich zum Löschen einer Kontoverbindung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                                       | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| DisassociateIamRoleFromConnection                  | Erforderlich, um eine IAM-Rolle von einer Kontoverbindung zu trennen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                 | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| GetBillingAuthorization                            | Erforderlich zum Beschreiben der Fakturierungsautorisierung für eine Kontoverbindung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| GetConnection                                      | Erforderlich zum Abrufen einer Kontoverbindung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                                       | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

| CodeCatalyst -Berechtigungen für Kontoverbindungen | Erforderliche Berechtigungen   | Ressourcen   |
|--|--|--|
| GetPendingConnection                               | Erforderlich, um eine ausstehende Anfrage zum Verbinden dieses Kontos mit einem CodeCatalyst Bereich zu erhalten. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | Unterstützt nur einen Platzhalter (*) im Resource-Richtlinienelement.              |
| ListConnections                                    | Erforderlich zum Auflisten von Kontoverbindungen, die nicht ausstehen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.  | Unterstützt nur einen Platzhalter (*) im Resource-Richtlinienelement.              |
| ListIamRolesForConnection                          | Erforderlich zum Auflisten von IAM-Rollen, die einer Kontoverbindung zugeordnet sind. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                             | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| ListTagsForResource                                | Erforderlich zum Auflisten von Tags, die einer Kontoverbindung zugeordnet sind. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                                   | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| PutBillingAuthorization                            | Erforderlich zum Erstellen oder Aktualisieren der Fakturierungsautorisierung für eine Kontoverbindung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.            | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

| CodeCatalyst -Berechtigungen für Kontoverbindungen | Erforderliche Berechtigungen  | Ressourcen   |
|--|---|--|
| RejectConnection                                   | Erforderlich, um eine Anforderung zum Verbinden dieses Kontos mit einem CodeCatalyst Bereich abzulehnen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | Unterstützt nur einen Platzhalter (*) im Resource-Richtlinienelement.              |
| TagResource  | Erforderlich zum Erstellen oder Bearbeiten von Tags, die einer Kontoverbindung zugeordnet sind. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.          | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| UntagResource                                      | Erforderlich zum Entfernen von Tags, die einer Kontoverbindung zugeordnet sind. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                          | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

## Erforderliche Berechtigungen für IAM-Identity-Center-Anwendungen

Die folgenden Berechtigungen sind für die Arbeit mit IAM-Identity-Center-Anwendungen erforderlich.

| CodeCatalyst -Berechtigungen für IAM-Identity-Center-Anwendungen | Erforderliche Berechtigungen  | Ressourcen   |
|--|---|--|
| AssociateIdentityCenterApplicationToSpace                        | Erforderlich, um eine IAM-Identity-Center-Anwendung mit einem CodeCatalyst Leerzeichen zu verknüpfen. Dies ist nur eine IAM-Richt | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applicati |

| CodeCatalyst -Berechtigungen für IAM-Identity-Center-Anwendungen | Erforderliche Berechtigungen   | Ressourcen   |
|--|--|--|
|  | linienberechtigung, keine API-Aktion.  | ons/ <i>identity-center-application_ID</i>   |
| AssociateIdentityToIdentityCenterApplication                     | Erforderlich, um eine Identität einer IAM-Identity-Center-Anwendung für einen CodeCatalyst Bereich zuzuordnen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                        | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| BatchAssociateIdentitiesToIdentityCenterApplication              | Erforderlich, um mehrere Identitäten einer IAM-Identity-Center-Anwendung für einen CodeCatalyst Bereich zuzuordnen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                   | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| BatchDisassociateIdentitiesFromIdentityCenterApplication         | Erforderlich, um die Zuordnung mehrerer Identitäten zu einer IAM-Identity-Center-Anwendung für einen CodeCatalyst Bereich aufzuheben. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| CreateIdentityCenterApplication                                  | Erforderlich zum Erstellen einer IAM-Identity-Center-Anwendung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.   | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst -Berechtigungen für IAM-Identity-Center-Anwendungen | Erforderliche Berechtigungen   | Ressourcen   |
|--|--|--|
| CreateSpaceAdminRoleAssignment                                   | Erforderlich zum Erstellen einer Administratorrolle zuweisung für einen bestimmten CodeCatalyst Bereich und eine IAM-Identity-Center-Anwendung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DeleteIdentityCenterApplication                                  | Erforderlich zum Löschen einer IAM-Identity-Center-Anwendung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.   | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DisassociateIdentityCenterApplicationFromSpace                   | Erforderlich, um die Zuordnung einer IAM-Identity-Center-Anwendung zu einem CodeCatalyst Bereich aufzuheben. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                                    | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DisassociateIdentityFromIdentityCenterApplication                | Erforderlich, um die Zuordnung einer Identität zu einer IAM-Identity-Center-Anwendung für einen CodeCatalyst Bereich aufzuheben. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst -Berechtigungen für IAM-Identity-Center-Anwendungen | Erforderliche Berechtigungen   | Ressourcen   |
|--|--|--|
| GetIdentityCenterApplication                                     | Erforderlich, um Informationen zu einer IAM-Identity-Center-Anwendung abzurufen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                            | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| ListIdentityCenterApplications                                   | Erforderlich, um eine Liste aller IAM-Identity-Center-Anwendungen im Konto anzuzeigen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                      | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| ListIdentityCenterApplicationsForSpace                           | Erforderlich, um eine Liste von IAM-Identity-Center-Anwendungen nach CodeCatalyst Speicherplatz anzuzeigen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| ListSpacesForIdentityCenterApplication                           | Erforderlich, um eine Liste von CodeCatalyst Leerzeichen nach IAM-Identity-Center-Anwendung anzuzeigen. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.     | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |



| CodeCatalyst -Berechtigungen für IAM-Identity-Center-Anwendungen | Erforderliche Berechtigungen  | Ressourcen   |
|--|---|--|
| SynchronizIdentityCenterApplication                              | Erforderlich zum Synchronisieren einer IAM-Identity-Center-Anwendung mit dem Backing-Identitätsspeicher. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| UpdateIdentityCenterApplication                                  | Erforderlich zum Aktualisieren einer IAM-Identity-Center-Anwendung. Dies ist nur eine IAM-Richtlinienberechtigung, keine API-Aktion.                                      | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

## Verwenden von serviceverknüpften Rollen für CodeCatalyst

Amazon CodeCatalyst verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit CodeCatalyst verknüpft ist. Serviceverknüpfte Rollen werden von CodeCatalyst vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer -AWSServices in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle vereinfacht die Einrichtung von CodeCatalyst, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. CodeCatalyst definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt wurde, kann CodeCatalyst nur die Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre CodeCatalyst Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceorientierte Rollen unterstützen, finden Sie unter [AWS services that work with IAM](#) (-Services, die mit IAM funktionieren). Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Service-linked roles (Serviceorientierte Rollen) angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

## Berechtigungen von serviceverknüpften Rollen für CodeCatalyst

CodeCatalyst verwendet die serviceverknüpfte Rolle namens `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` – Ermöglicht Amazon CodeCatalyst schreibgeschützten Zugriff auf Anwendungs-Instance-Profile und zugehörige Verzeichnisbenutzer und -gruppen in Ihrem Namen.

### Die serviceverknüpfte Rolle

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` vertraut darauf, dass die folgenden Services die Rolle annehmen:

- `codecatalyst.amazonaws.com`

### Die Rollenberechtigungsrichtlinie namens

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy` ermöglicht es CodeCatalyst, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `View application instance profiles and associated directory users and groups` für CodeCatalyst spaces that support identity federation and SSO users and groups

Sie müssen Berechtigungen konfigurieren, damit eine Benutzer, Gruppen oder Rollen eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter [serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

## Erstellen einer serviceverknüpften Rolle für CodeCatalyst

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie einen Bereich in der AWS CLI, AWS Management Console oder der -AWSAPI erstellen, CodeCatalyst erstellt die serviceverknüpfte Rolle für Sie.

**⚠ Important**

Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Features verwendet. Wenn Sie den CodeCatalyst Service vor dem 17. November 2023 verwendet haben, als dieser serviceverknüpfte Rollen unterstützte, dann hat die `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` Rolle in Ihrem Konto CodeCatalyst erstellt. Weitere Informationen finden Sie unter [In meinem AWS-Konto](#) wird eine neue Rolle angezeigt.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Bereich erstellen, CodeCatalyst erstellt die serviceverknüpfte Rolle erneut für Sie.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall Anwendungs-Instance-Profile anzeigen und zugehörigen Verzeichnisbenutzern und -gruppen zu erstellen. Erstellen Sie in der AWS CLI oder der AWS-API eine serviceverknüpfte Rolle mit dem Servicenamen `codecatalyst.amazonaws.com`. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

### Bearbeiten einer serviceverknüpften Rolle für CodeCatalyst

CodeCatalyst erlaubt Ihnen nicht, die `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

### Löschen einer serviceverknüpften Rolle für CodeCatalyst

Sie müssen die Rolle `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` nicht manuell löschen. Wenn Sie ein Leerzeichen in der AWS CLI, AWS Management Console oder der API löschenAWS, CodeCatalyst bereinigt die Ressourcen und löscht die serviceverknüpfte Rolle für Sie.

Sie können jedoch auch die IAM-Konsole, AWS CLI oder AWS API verwenden, um eine serviceverknüpfte Rolle manuell zu löschen. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zuerst manuell bereinigen, bevor Sie diese manuell löschen können.

### Note

Wenn der CodeCatalyst Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie CodeCatalyst Ressourcen, die von der verwendet werden  
AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization

- [Löschen Sie das Leerzeichen](#).

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, AWS CLI- oder AWS-API, um die AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für CodeCatalyst serviceverknüpfte Rollen

CodeCatalyst unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS Regionen und Endpunkte](#).

CodeCatalyst unterstützt nicht die Verwendung von serviceverknüpften Rollen in jeder Region, in der der Service verfügbar ist. Sie können die Rolle AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization in den folgenden Regionen verwenden.

| Name der Region         | Regions-ID | Support in CodeCatalyst |
|-------------------------|------------|-------------------------|
| USA Ost (Nord-Virginia) | us-east-1  | Nein                    |
| USA Ost (Ohio)          | us-east-2  | Nein                    |

| Name der Region            | Regions-ID     | Support in CodeCatalyst |
|----------------------------|----------------|-------------------------|
| USA West (Nordkalifornien) | us-west-1      | Nein                    |
| USA West (Oregon)          | us-west-2      | Ja                      |
| Afrika (Kapstadt)          | af-south-1     | Nein                    |
| Asien-Pazifik (Hongkong)   | ap-east-1      | Nein                    |
| Asien-Pazifik (Jakarta)    | ap-southeast-3 | Nein                    |
| Asien-Pazifik (Mumbai)     | ap-south-1     | Nein                    |
| Asien-Pazifik (Osaka)      | ap-northeast-3 | Nein                    |
| Asien-Pazifik (Seoul)      | ap-northeast-2 | Nein                    |
| Asien-Pazifik (Singapur)   | ap-southeast-1 | Nein                    |
| Asien-Pazifik (Sydney)     | ap-southeast-2 | Nein                    |
| Asien-Pazifik (Tokio)      | ap-northeast-1 | Nein                    |
| Kanada (Zentral)           | ca-central-1   | Nein                    |
| Europa (Frankfurt)         | eu-central-1   | Nein                    |
| Europa (Irland)            | eu-west-1      | Ja                      |
| Europa (London)            | eu-west-2      | Nein                    |
| Europa (Mailand)           | eu-south-1     | Nein                    |
| Europa (Paris)             | eu-west-3      | Nein                    |
| Europa (Stockholm)         | eu-north-1     | Nein                    |
| Naher Osten (Bahrain)      | me-south-1     | Nein                    |
| Naher Osten (VAE)          | me-central-1   | Nein                    |

| Name der Region           | Regions-ID    | Support in CodeCatalyst |
|---------------------------|---------------|-------------------------|
| South America (São Paulo) | sa-east-1     | Nein                    |
| AWS GovCloud (USA-Ost)    | us-gov-east-1 | Nein                    |
| AWS GovCloud (USA-West)   | us-gov-west-1 | Nein                    |

## AWS Von verwaltete Richtlinien für Amazon CodeCatalyst

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Services verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

### AWS verwaltete Richtlinie: AmazonCodeCatalystSupportAccess

Dies ist eine Richtlinie, die allen Raumadministratoren und Raummitgliedern Berechtigungen erteilt, den Premium-Supportplan Business oder Enterprise zu nutzen, der dem Raumabrechnungskonto

zugeordnet ist. Diese Berechtigungen ermöglichen es Raumadministratoren und Mitgliedern, den Premium-Supportplan für die Ressourcen, für die sie Berechtigungen haben, innerhalb von CodeCatalyst Berechtigungsrichtlinien zu nutzen.

## Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- **support** – Gewährt Berechtigungen, damit Benutzer Support-AWSFälle suchen, erstellen und lösen können. Gewährt auch Berechtigungen zum Beschreiben von Kommunikation, Schweregraden, Anhängen und zugehörigen Support-Falldetails.

```
{
  "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 verwaltete Richtlinie: AmazonCodeCatalystFullAccess

Dies ist eine Richtlinie, die Berechtigungen zum Verwalten Ihres CodeCatalyst Bereichs und Ihrer verbundenen Konten auf der Amazon- CodeCatalyst Spaces-Seite in der gewährtAWS Management Console. Diese Anwendung wird verwendet, um zu konfigurierenAWS-Konten, die mit Ihrem Speicherplatz in verbunden sind CodeCatalyst.

### Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `codecatalyst` – Gewährt vollständige Berechtigungen für die Seite Amazon CodeCatalyst Spaces im AWS Management Console.

```
{
  "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 verwaltete Richtlinie: AmazonCodeCatalystReadOnlyAccess

Dies ist eine Richtlinie, die Berechtigungen zum Anzeigen und Auflisten von Informationen für Leerzeichen und verbundene Konten auf der Seite Amazon CodeCatalyst Spaces in der AWS Management Console gewährt. Diese Anwendung wird verwendet, um zu konfigurieren AWS-Konten, die mit Ihrem Speicherplatz in verbunden sind CodeCatalyst.

#### Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `codecatalyst` – Gewährt Leseberechtigungen für die Seite Amazon CodeCatalyst Spaces in der AWS Management Console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:Get*",
        "codecatalyst:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS verwaltete Richtlinie:

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy

Sie können nicht an Ihre

IAMAmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy-Entitäten anfügen. Diese Richtlinie ist an eine serviceverknüpfte Rolle angehängt, die es ermöglicht CodeCatalyst , Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter [Verwenden von serviceverknüpften Rollen für CodeCatalyst](#).

Diese Richtlinie ermöglicht es Kunden, Anwendungs-Instance-Profile und zugehörige Verzeichnisbenutzer und -gruppen anzuzeigen, wenn sie Bereiche in verwalteten CodeCatalyst. Kunden werden diese Ressourcen anzeigen, wenn sie Bereiche verwalten, die den Identitätsverbund und SSO-Benutzer und -Gruppen unterstützen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- sso – Gewährt Berechtigungen, damit Benutzer Anwendungs-Instance-Profile anzeigen können, die in IAM Identity Center für zugehörige Bereiche in verwaltet werden CodeCatalyst.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy",
      "Effect": "Allow",
      "Action": [
        "sso:ListInstances",
        "sso:ListApplications",
        "sso:ListApplicationAssignments",
        "sso:DescribeInstance",
        "sso:DescribeApplication"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

## CodeCatalyst -Aktualisierungen für -AWSverwaltete Richtlinien

Zeigen Sie Details zu Aktualisierungen für -AWSverwaltete Richtlinien für an, CodeCatalyst seit dieser Service mit der Verfolgung dieser Änderungen begonnen hat. Um automatische Warnungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS-Feed auf der Seite CodeCatalyst [Dokumentverlauf](#).

| Änderung  | Beschreibung   | Datum             |
|---|--|-------------------|
| <a href="#">AmazonCodeCatalyst ServiceRoleForIdentityCenterApplicationSynchronization Policy</a> – Neue Richtlinie. | CodeCatalyst hat die Richtlinie hinzugefügt.<br><br>Gewährt Berechtigungen, damit CodeCatalyst Benutzer Anwendungs-Instance-Profile und zugehörige Verzeichnissebenutzer und -gruppen anzeigen können.       | 17. November 2023 |
| <a href="#">AmazonCodeCatalyst SupportAccess</a> – Neue Richtlinie.   | CodeCatalyst hat die Richtlinie hinzugefügt.<br><br>Gewährt Berechtigungen, damit CodeCatalyst Benutzer Support-Fälle suchen, erstellen und lösen sowie zugehörige Mitteilungen und Details anzeigen können. | 20. April 2023    |
| <a href="#">AmazonCodeCatalyst FullAccess</a> – Neue Richtlinie.  | CodeCatalyst hat die Richtlinie hinzugefügt.<br><br>Gewährt vollen Zugriff auf CodeCatalyst.   | 20. April 2023    |

| Änderung   | Beschreibung   | Datum          |
|--|--|----------------|
| <a href="#">AmazonCodeCatalyst ReadOnlyAccess</a> – Neue Richtlinie. | CodeCatalyst hat die Richtlinie hinzugefügt.<br><br>Gewährt schreibgeschützten Zugriff auf CodeCatalyst. | 20. April 2023 |
| CodeCatalyst hat mit der Verfolgung von Änderungen begonnen          | CodeCatalyst hat mit der Verfolgung von Änderungen für seine AWS -verwalteten Richtlinien begonnen.      | 20. April 2023 |

## IAM-Rollen für Amazon- CodeCatalyst Zugriff auf -AWSRessourcen

CodeCatalyst kann auf -AWSRessourcen zugreifen, indem Sie Ihre AWS-Konto mit einem CodeCatalyst Bereich verbinden. Sie können dann die folgenden Servicerollen erstellen und sie verknüpfen, wenn Sie Ihr Konto verbinden.

Weitere Informationen zu den Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie unter [IAM-JSON-Richtlinienelementreferenz](#) im IAM-Benutzerhandbuch.

- Um auf Ressourcen in einem AWS-Konto für Ihre CodeCatalyst Projekte und Workflows zuzugreifen, müssen Sie zunächst die Berechtigung für erteilen CodeCatalyst , in Ihrem Namen auf diese Ressourcen zuzugreifen. Dazu müssen Sie eine Servicerolle in einem verbundenen erstellenAWS-Konto, die CodeCatalyst im Namen von Benutzern und Projekten im Bereich übernehmen kann. Sie können entweder die CodeCatalystWorkflowDevelopmentRole-**spaceName** Servicerolle erstellen und verwenden oder benutzerdefinierte Servicerollen erstellen und diese IAM-Richtlinien und -Rollen manuell konfigurieren. Als bewährte Methode sollten Sie diesen Rollen die geringsten erforderlichen Berechtigungen zuweisen.

### Note

Für benutzerdefinierte Servicerollen ist der CodeCatalyst Service-Prinzipal erforderlich. Weitere Informationen zum CodeCatalyst Service-Prinzipal und zum Vertrauensmodell finden Sie unter [CodeCatalyst Vertrauensmodell](#).

- Um den Support für einen Bereich über die verbundenen zu verwalten AWS-Konto, können Sie die `AWSRoleForCodeCatalystSupport` Servicerolle erstellen und verwenden, mit der CodeCatalyst Benutzer auf den Support zugreifen können. Weitere Informationen zur Unterstützung eines CodeCatalyst Bereichs finden Sie unter [AWS Support für Amazon CodeCatalyst](#).

## Grundlegendes zur `CodeCatalystWorkflowDevelopmentRole-spaceName` Servicerolle

Sie können eine IAM-Rolle für Ihren Bereich hinzufügen, die verwenden CodeCatalyst kann, um Ressourcen in einem verbundenen zu erstellen und darauf zuzugreifen AWS-Konto. Dies wird als [Servicerolle](#) bezeichnet. Die einfachste Möglichkeit, eine Servicerolle zu erstellen, besteht darin, eine hinzuzufügen, wenn Sie den Speicherplatz erstellen, und die `CodeCatalystWorkflowDevelopmentRole-spaceName` Option für diese Rolle auszuwählen. Dadurch wird nicht nur die Servicerolle mit dem `AdministratorAccess` angehängten erstellt, sondern auch die Vertrauensrichtlinie, die es ermöglicht CodeCatalyst, die Rolle im Namen von Benutzern in Projekten im Raum zu übernehmen. Die Servicerolle ist auf den Bereich beschränkt, nicht auf einzelne Projekte. Informationen zum Erstellen dieser Rolle finden Sie unter [Erstellen der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle für Ihr Konto und Ihren Bereich](#). Sie können nur eine Rolle für jeden Bereich in jedem Konto erstellen.

### Note

Diese Rolle wird nur für die Verwendung mit -Entwicklungskonten empfohlen und verwendet die `-AdministratorAccess` AWS-verwaltete Richtlinie, sodass sie vollen Zugriff auf die Erstellung neuer Richtlinien und Ressourcen in dieser hat AWS-Konto.

Die der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle angefügte Richtlinie ist so konzipiert, dass sie mit Projekten funktioniert, die mit Blueprints im Raum erstellt wurden. Es ermöglicht Benutzern in diesen Projekten, Code mithilfe von Ressourcen in der verbundenen zu entwickeln, zu erstellen, zu testen und bereitzustellen AWS-Konto. Weitere Informationen finden Sie unter [Erstellen einer Rolle für einen -AWS-Service](#).

Die der `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle angefügte Richtlinie ist die `AdministratorAccess` verwaltete Richtlinie in AWS. Dies ist eine Richtlinie, die vollen Zugriff auf alle AWS Aktionen und Ressourcen gewährt. Informationen zum Anzeigen des JSON-Richtliniendokuments in der IAM-Konsole finden Sie unter [AdministratorAccess](#).

Mit der folgenden Vertrauensrichtlinie kann die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle CodeCatalyst übernehmen. Weitere Informationen zum CodeCatalyst Vertrauensmodell finden Sie unter [CodeCatalyst Vertrauensmodell](#).

```
"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/*"
        }
      }
    }
  ]
```

Erstellen der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle für Ihr Konto und Ihren Bereich

Gehen Sie wie folgt vor, um die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle zu erstellen, die für Workflows in Ihrem Bereich verwendet wird. Für jedes Konto, das Sie über IAM-Rollen für die Verwendung in Projekten verfügen möchten, müssen Sie Ihrem Bereich eine Rolle wie die Entwicklerrolle hinzufügen.


Bevor Sie beginnen, müssen Sie über Administratorrechte für Ihr verfügbares AWS-Konto oder mit Ihrem Administrator zusammenarbeiten können. Weitere Informationen zur Verwendung von AWS-Konten und IAM- CodeCatalystRollen in finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

So erstellen Sie die und fügen sie hinzu CodeCatalyst  
CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die und stellen Sie sicherAWS Management Console, dass Sie AWS-Konto für Ihren Speicherplatz mit derselben angemeldet sind.
2. Öffnen Sie die - CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
3. Navigieren Sie zu Ihrem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
4. Wählen Sie den Link für die ausAWS-Konto, in der Sie die Rolle erstellen möchten. Die AWS-Konto Detailseite wird angezeigt.
5. Wählen Sie Rollen verwalten ausAWS Management Console.

Die Seite IAM-Rolle zum Amazon- CodeCatalyst Bereich hinzufügen wird im geöffnetAWS Management Console. Dies ist die Seite Amazon CodeCatalyst Spaces. Möglicherweise müssen Sie sich anmelden, um auf die Seite zugreifen zu können.

6. Wählen Sie CodeCatalyst Entwicklungsadministratorrolle in IAM erstellen aus. Diese Option erstellt eine Servicerolle, die die Berechtigungsrichtlinie und Vertrauensrichtlinie für die Entwicklungsrolle enthält. Die Rolle hat den Namen CodeCatalystWorkflowDevelopmentRole-*spaceName*. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#).

 Note

Diese Rolle wird nur für die Verwendung mit Entwicklerkonten empfohlen und verwendet die von AdministratorAccess AWS verwaltete Richtlinie, sodass sie vollen Zugriff auf die Erstellung neuer Richtlinien und Ressourcen in dieser hatAWS-Konto.

7. Wählen Sie Entwicklungsrolle erstellen aus.
8. Zeigen Sie auf der Seite Verbindungen unter IAM-Rollen, die für verfügbar sind CodeCatalyst die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle in der Liste der IAM-Rollen an, die Ihrem Konto hinzugefügt wurden.
9. Um zu Ihrem Bereich zurückzukehren, wählen Sie Zu Amazon gehen aus CodeCatalyst.

## Grundlegendes zur AWSRoleForCodeCatalystSupport Servicerolle

Sie können eine IAM-Rolle für Ihren Bereich hinzufügen, mit der CodeCatalyst Benutzer in einem Bereich Support-Fälle erstellen und darauf zugreifen können. Dies wird als [Servicerolle](#) für den Support bezeichnet. Am einfachsten können Sie eine Servicerolle für den Support erstellen, indem Sie eine hinzufügen, wenn Sie den Speicherplatz erstellen, und die `AWSRoleForCodeCatalystSupport` Option für diese Rolle auswählen. Dadurch werden nicht nur die Richtlinie und die Rolle erstellt, sondern auch die Vertrauensrichtlinie, die es ermöglicht CodeCatalyst, die Rolle im Namen von Benutzern in Projekten im Raum zu übernehmen. Die Servicerolle ist auf den Bereich beschränkt, nicht auf einzelne Projekte. Informationen zum Erstellen dieser Rolle finden Sie unter [Erstellen der AWSRoleForCodeCatalystSupport Rolle für Ihr Konto und Ihren Bereich](#).

Die der `AWSRoleForCodeCatalystSupport` Rolle angefügte Richtlinie ist eine verwaltete Richtlinie, die Zugriff auf Support-Berechtigungen bietet. Weitere Informationen finden Sie unter [AWS verwaltete Richtlinie: AmazonCodeCatalystSupportAccess](#).

Die Vertrauensrolle für die Richtlinie erlaubt CodeCatalyst, die Rolle zu übernehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst.amazonaws.com",
          "codecatalyst-runner.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

### Erstellen der AWSRoleForCodeCatalystSupport Rolle für Ihr Konto und Ihren Bereich

Gehen Sie wie folgt vor, um die `AWSRoleForCodeCatalystSupport` Rolle zu erstellen, die für Support-Fälle in Ihrem Bereich verwendet wird. Die Rolle muss dem angegebenen Fakturierungskonto für den Speicherplatz hinzugefügt werden.



Bevor Sie beginnen, müssen Sie über Administratorrechte für Ihr verfügbares AWS-Konto oder mit Ihrem Administrator zusammenarbeiten können. Weitere Informationen zur Verwendung von AWS-Konten und IAM- CodeCatalystRollen in finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

So erstellen Sie die und fügen sie hinzu CodeCatalyst AWSRoleForCodeCatalystSupport

1. Bevor Sie in der CodeCatalyst Konsole beginnen, öffnen Sie die und stellen Sie sicher, dass Sie sich angemeldet sind.
2. Navigieren Sie zu Ihrem CodeCatalyst Bereich. Wählen Sie Settings (Einstellungen) und anschließend AWS-Konten aus.
3. Wählen Sie den Link für die ausAWS-Konto, in der Sie die Rolle erstellen möchten. Die AWS-Konto Detailseite wird angezeigt.
4. Wählen Sie Rollen verwalten ausAWS Management Console.

Die Seite IAM-Rolle zum Amazon- CodeCatalyst Bereich hinzufügen wird im geöffneten AWS Management Console. Dies ist die Seite Amazon CodeCatalyst Spaces. Möglicherweise müssen Sie sich anmelden, um auf die Seite zugreifen zu können.

5. Wählen Sie unter CodeCatalyst Raumdetails die Option CodeCatalyst Supportrolle hinzufügen aus. Diese Option erstellt eine Servicerolle, die die Berechtigungsrichtlinie und Vertrauensrichtlinie für die Vorschau-Entwicklungsrolle enthält. Der Rolle wird ein Name `AWSRoleForCodeCatalystSupport` mit einer eindeutigen Kennung angehängt. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur AWSRoleForCodeCatalystSupport Servicerolle](#).
6. Lassen Sie auf der Seite Rolle für CodeCatalyst Support hinzufügen die Standardeinstellung ausgewählt und wählen Sie dann Rolle erstellen aus.
7. Zeigen Sie unter IAM-Rollen, die für verfügbar sind CodeCatalyst, die `CodeCatalystWorkflowDevelopmentRole-spaceName` Rolle in der Liste der IAM-Rollen an, die Ihrem Konto hinzugefügt wurden.
8. Um zu Ihrem Bereich zurückzukehren, wählen Sie Zu Amazon gehen aus CodeCatalyst.

## Konfigurieren von IAM-Rollen für Workflow-Aktionen in CodeCatalyst

In diesem Abschnitt werden IAM-Rollen und -Richtlinien beschrieben, die Sie zur Verwendung mit Ihrem CodeCatalyst Konto erstellen können. Anweisungen zum Erstellen von Beispielrollen finden Sie unter [Manuelles Erstellen von Rollen für Workflow-Aktionen](#). Nachdem Sie Ihre IAM-Rolle erstellt

haben, kopieren Sie den Rollen-ARN, um die IAM-Rolle zu Ihrer Kontoverbindung hinzuzufügen und sie Ihrer Projektumgebung zuzuordnen. Weitere Informationen hierzu finden Sie unter [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#).

### CodeCatalyst -Build-Rolle für Amazon S3-Zugriff

Für CodeCatalyst Workflow-Build-Aktionen können Sie die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*ServiceRolle verwenden oder eine IAM-Rolle mit dem Namen CodeCatalystBuildRoleforS3Access erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf AWS CloudFormation Ressourcen in Ihrem CodeCatalyst benötigtAWS-Konto.

Diese Rolle gewährt Berechtigungen für Folgendes:

- Schreiben in Amazon S3-Buckets.
- Unterstützung für die Erstellung von -Ressourcen mit AWS CloudFormation. Dies erfordert Amazon S3-Zugriff.

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  }]
}
```

#### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach ihrer Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst -Build-Rolle für AWS CloudFormation

Für CodeCatalyst Workflow-Build-Aktionen können Sie die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*Service-Rolle verwenden oder eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben für AWS CloudFormation Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Unterstützung für die Erstellung von -Ressourcen mit AWS CloudFormation. Dies ist zusammen mit der CodeCatalyst Build-Rolle für den Amazon S3-Zugriff und der CodeCatalyst Bereitstellungsrolle für erforderlich AWS CloudFormation.

Die folgenden AWS verwalteten Richtlinien sollten an diese Rolle angehängt werden:

- AWSCloudFormationFullAccess
- IAMFullAccess
- AmazonS3FullAccess
- AmazonAPIGatewayAdministrator
- AWSLambdaFullAccess

## CodeCatalyst Build-Rolle für CDK

Für CodeCatalyst Workflows, die CDK-Build-Aktionen ausführen, wie z. B. die moderne dreistufige Webanwendung, können Sie die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*Service-Rolle verwenden oder eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Bootstrappen und Ausführen von CDK-Build-Befehlen für AWS CloudFormation Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Schreiben in Amazon S3-Buckets.
- Unterstützung für die Erstellung von CDK-Konstrukten und AWS CloudFormation Ressourcen-Stacks. Dies erfordert den Zugriff auf Amazon S3 für Artefaktspeicher, Amazon ECR für Image-Repository-Unterstützung und SSM für System-Governance und Überwachung für virtuelle Instances.

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst Rolle für bereitstellen AWS CloudFormation

Für CodeCatalyst Workflow-Bereitstellungsaktionen, die verwenden AWS CloudFormation, können Sie die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*ServiceRolle oder eine Richtlinie mit bereichsbezogenen Berechtigungen verwenden, die zum Ausführen von Aufgaben für AWS CloudFormation Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Zulassen CodeCatalyst , dass eine aufruft, um eine Blau/Grün-Bereitstellung über durchzuführenAWS CloudFormation.
- Erlauben Sie CodeCatalyst das Erstellen und Aktualisieren von Stacks und Änderungssätzen in AWS CloudFormation.

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "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

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst Rolle für Amazon EC2 bereitstellen

CodeCatalyst -Workflow-Bereitstellungsaktionen verwenden eine IAM-Rolle mit den erforderlichen Berechtigungen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf Amazon EC2-Ressourcen in Ihrem CodeCatalyst benötigtAWS-

Konto. Die Standardrichtlinie für die CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle enthält keine Berechtigungen für Amazon EC2 oder Amazon EC2 Auto Scaling .

Diese Rolle erteilt Berechtigungen für Folgendes:

- Erstellen Sie Amazon EC2-Bereitstellungen.
- Lesen Sie die Tags auf einer Instance oder identifizieren Sie eine Amazon EC2-Instance anhand der Auto Scaling-Gruppenamen.
- Lesen, erstellen, aktualisieren und löschen Sie Amazon EC2 Auto Scaling-Gruppen, Lebenszyklus-Hooks und Skalierungsrichtlinien.
- Veröffentlichen Sie Informationen zu Amazon SNS-Themen.
- Abrufen von Informationen zu CloudWatch Alarmen.
- Lesen und aktualisieren Sie Elastic Load Balancing .

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction",
        "autoscaling>DeleteLifecycleHook",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:PutLifecycleHook",
        "autoscaling:RecordLifecycleActionHeartbeat",
        "autoscaling>CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:EnableMetricsCollection",
        "autoscaling:DescribePolicies",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeNotificationConfigurations",
        "autoscaling:SuspendProcesses",
        "autoscaling:ResumeProcesses",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:AttachLoadBalancerTargetGroups",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
```

```

"autoscaling:PutNotificationConfiguration",
"autoscaling:PutWarmPool",
"autoscaling:DescribeScalingActivities",
"autoscaling>DeleteAutoScalingGroup",
"ec2:DescribeInstances",
"ec2:DescribeInstanceStatus",
"ec2:TerminateInstances",
"tag:GetResources",
"sns:Publish",
"cloudwatch:DescribeAlarms",
"cloudwatch:PutMetricAlarm",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeInstanceHealth",
"elasticloadbalancing:RegisterInstancesWithLoadBalancer",
"elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:DescribeTargetHealth",
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DeregisterTargets"
],
"Resource": "resource_ARN"
  }
]
}

```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst Rolle für Amazon ECS bereitstellen

Für CodeCatalyst Workflow-Aktionen können Sie eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Sie können die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*Service-Rolle verwenden oder eine IAM-Rolle für CodeCatalyst Bereitstellungsaktionen erstellen, die für Lambda-Bereitstellungen verwendet werden sollen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen

Berechtigungen, die zum Ausführen von Aufgaben auf Amazon-ECS-Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Initiieren Sie die fortlaufende Amazon-ECS-Bereitstellung im Namen eines CodeCatalyst Benutzers in einem Konto, das in der CodeCatalyst Verbindung angegeben ist.
- Lesen, aktualisieren und löschen Sie Amazon-ECS-Aufgabensätze.
- Aktualisieren Sie Zielgruppen, Listener und Regeln von Elastic Load Balancing.
- Rufen Sie Lambda-Funktionen auf.
- Greifen Sie auf Revisionsdateien in Amazon S3-Buckets zu.
- Abrufen von Informationen zu CloudWatch Alarmen.
- Veröffentlichen Sie Informationen zu Amazon SNS-Themen.

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "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

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"
```

## CodeCatalyst Rolle für Lambda bereitstellen

Für CodeCatalyst Workflow-Aktionen können Sie eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Sie können die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*ServiceRolle verwenden oder eine IAM-Rolle für CodeCatalyst Bereitstellungsaktionen erstellen, die für Lambda-Bereitstellungen verwendet werden sollen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf Lambda-Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Lesen, aktualisieren und rufen Sie Lambda-Funktionen und -Aliase auf.
- Greifen Sie auf Revisionsdateien in Amazon S3-Buckets zu.
- Abrufen von Informationen zu CloudWatch Ereignisalarmen.
- Veröffentlichen Sie Informationen zu Amazon SNS-Themen.

Diese Rolle verwendet die folgende Richtlinie:

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
```

```
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
        "StringEquals": {
            "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
    },
    "Effect": "Allow"
},
{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda:::function:CodeDeployHook_*",
    "Effect": "Allow"
}
]
}
```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst Rolle für Lambda bereitstellen

Für CodeCatalyst Workflow-Aktionen können Sie die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*Service-Rolle verwenden oder eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf Lambda-Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Lesen, aktualisieren und rufen Sie Lambda-Funktionen und -Aliase auf.
- Greifen Sie auf Revisionsdateien in Amazon S3-Buckets zu.
- Abrufen von Informationen zu CloudWatch Alarmen.
- Veröffentlichen Sie Informationen zu Amazon SNS-Themen.

Diese Rolle verwendet die folgende Richtlinie:

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::/CodeDeploy/",
      "Effect": "Allow"
    },
    {
      "Action": [
```

```

        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
        "StringEquals": {
            "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
    },
    "Effect": "Allow"
},
{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda::function:CodeDeployHook_*",
    "Effect": "Allow"
}
]
}

```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst Rolle für bereitstellen AWS SAM

Für CodeCatalyst Workflow-Aktionen können Sie die Standard-CodeCatalystWorkflowDevelopmentRole-*spaceName*ServiceRolle verwenden oder eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf AWS SAM und AWS CloudFormation Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Zulassen CodeCatalyst, dass eine Lambda-Funktion aufruft, um die Bereitstellung von Serverless- und AWS SAM CLI-Anwendungen durchzuführen.
- Erlauben Sie CodeCatalyst das Erstellen und Aktualisieren von Stacks und Änderungssätzen in AWS CloudFormation.

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst schreibgeschützte Rolle für Amazon EC2

Für CodeCatalyst Workflow-Aktionen können Sie eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf Amazon EC2-Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto. Die CodeCatalystWorkflowDevelopmentRole-*spaceName* Servicerolle enthält keine Berechtigungen für Amazon EC2 oder die beschriebenen Aktionen für Amazon CloudWatch.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Rufen Sie den Status von Amazon EC2-Instances ab.
- Rufen Sie CloudWatch Metriken für Amazon EC2-Instances ab.

Diese Rolle verwendet die folgende Richtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe",
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe",
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:Describe"
      ],
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:Describe",
      "Resource": "resource_ARN"
    }
  ]
}
```

```

    }
  ]
}
```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## CodeCatalyst schreibgeschützte Rolle für Amazon ECS

Für CodeCatalyst Workflow-Aktionen können Sie eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf Amazon-ECS-Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Lesen Sie Amazon-ECS-Aufgabensätze.
- Abrufen von Informationen zu CloudWatch Alarmen.

Diese Rolle verwendet die folgende Richtlinie:

```

*{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ecs:DescribeServices",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [

```



```

        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DescribeRules"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
},
{
    "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
        "StringEquals": {
            "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
    },
    "Effect": "Allow"
},
{
    "Action": [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:iam::role/ecsTaskExecutionRole",
        "arn:aws:iam::role/ECSTaskExecution"
    ],
    "Condition": {
        "StringLike": {
            "iam:PassedToService": [
                "ecs-tasks.amazonaws.com"
            ]
        }
    }
}
]
}

```

**Note**

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"
```

## CodeCatalyst schreibgeschützte Rolle für Lambda

Für CodeCatalyst Workflow-Aktionen können Sie eine IAM-Rolle mit den erforderlichen Berechtigungen erstellen. Diese Rolle verwendet eine Richtlinie mit bereichsbezogenen Berechtigungen, die zum Ausführen von Aufgaben auf Lambda-Ressourcen in Ihrem CodeCatalyst benötigt AWS-Konto.

Diese Rolle erteilt Berechtigungen für Folgendes:

- Lesen Sie Lambda-Funktionen und -Aliase.
- Greifen Sie auf Revisionsdateien in Amazon S3-Buckets zu.
- Abrufen von Informationen zu CloudWatch Alarmen.

Diese Rolle verwendet die folgende -Richtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
      }
    },
    "Effect": "Allow"
  }
]
}

```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

## Manuelles Erstellen von Rollen für Workflow-Aktionen

CodeCatalyst -Workflow-Aktionen verwenden IAM-Rollen, die Sie erstellen und die als Build-Rolle, Bereitstellungsrolle und Stack-Rolle bezeichnet werden.

Gehen Sie wie folgt vor, um diese Rollen in IAM zu erstellen.

So erstellen Sie eine Bereitstellungsrolle

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich bei AWS an.

- b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- c. Wählen Sie im Navigationsbereich Policies aus.
- d. Wählen Sie Richtlinie erstellen aus.
- e. Wählen Sie den Tab JSON.
- f. Löschen Sie den vorhandenen Code.
- g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

#### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.

- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie unter Name Folgendes ein:

**codecatalyst-deploy-policy**

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Bereitstellungsrolle wie folgt:

- a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie aus.
- c. Löschen Sie die vorhandene benutzerdefinierte Vertrauensrichtlinie.
- d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach dem entsprechenden Kontrollkästchen **codecatalyst-deploy-policy** und aktivieren Sie es.
- g. Wählen Sie Weiter aus.
- h. Geben Sie für Rollenname Folgendes ein:

**codecatalyst-deploy-role**

- i. Geben Sie für Rollenbeschreibung Folgendes ein:

```
CodeCatalyst deploy role
```

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Bereitstellungsrolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Rufen Sie den ARN der Bereitstellungsrolle wie folgt ab:

- a. Wählen Sie im Navigationsbereich Rollen aus.
- b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (codecatalyst-deploy-role).
- c. Wählen Sie die Rolle aus der Liste aus.

Die Seite Zusammenfassung der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.

Sie haben jetzt die Bereitstellungsrolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

So erstellen Sie eine Build-Rolle

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:

- a. Melden Sie sich bei AWS an.
- b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- c. Wählen Sie im Navigationsbereich Policies aus.
- d. Wählen Sie Richtlinie erstellen aus.
- e. Wählen Sie den Tab JSON.
- f. Löschen Sie den vorhandenen Code.
- g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [{
```

```

    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}

```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie unter Name Folgendes ein:

```
codecatalyst-build-policy

```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

## 2. Erstellen Sie die Build-Rolle wie folgt:

- a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie aus.
- c. Löschen Sie die vorhandene benutzerdefinierte Vertrauensrichtlinie.
- d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
```

- e. Wählen Sie Weiter aus.
- f. Suchen Sie unter Berechtigungsrichtlinien nach dem entsprechenden Kontrollkästchen `codecatalyst-build-policy` und aktivieren Sie es.
- g. Wählen Sie Weiter aus.
- h. Geben Sie für Rollenname Folgendes ein:

**codecatalyst-build-role**

- i. Geben Sie für Rollenbeschreibung Folgendes ein:

**CodeCatalyst build role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Rufen Sie den ARN der Build-Rolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-build-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.


Die Seite Zusammenfassung der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.



Sie haben jetzt die Build-Rolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

So erstellen Sie eine Stack-Rolle

 Note

Sie müssen keine Stack-Rolle erstellen, obwohl dies aus Sicherheitsgründen empfohlen wird. Wenn Sie die Stack-Rolle nicht erstellen, müssen Sie die in diesem Verfahren beschriebenen Berechtigungsrichtlinien zur Bereitstellungsrolle hinzufügen.

1. Melden Sie sich bei AWS mit dem Konto an, in dem Sie Ihren Stack bereitstellen möchten.
2. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
3. Wählen Sie im Navigationsbereich Rollen und dann Rolle erstellen aus.
4. Wählen Sie oben AWS Service aus.
5. Wählen Sie in der Liste der Services die Option ausCloudFormation.
6. Wählen Sie Weiter: Berechtigungen aus.
7. Fügen Sie im Suchfeld alle Richtlinien hinzu, die für den Zugriff auf die Ressourcen in Ihrem Stack erforderlich sind. Wenn Ihr Stack beispielsweise eine -AWS LambdaFunktion enthält, müssen Sie eine Richtlinie hinzufügen, die Zugriff auf Lambda gewährt.

 Tip

Wenn Sie sich nicht sicher sind, welche Richtlinien hinzugefügt werden sollen, können Sie sie vorerst weglassen. Wenn Sie die Aktion testen und nicht über die richtigen Berechtigungen verfügen, AWS CloudFormation generiert Fehler, die zeigen, welche Berechtigungen Sie hinzufügen müssen.

8. Wählen Sie Next: Tags (Weiter: Tags) aus.
9. Klicken Sie auf Weiter: Prüfen.
10. Geben Sie für Rollename Folgendes ein:

**codecatalyst-stack-role**

11. Wählen Sie Rolle erstellen aus.
12. Gehen Sie wie folgt vor, um den ARN der Stack-Rolle abzurufen:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (codecatalyst-stack-role).
  - c. Wählen Sie die Rolle aus der Liste aus.
  - d. Kopieren Sie auf der Seite Zusammenfassung den Rollen-ARN-Wert.

## Verwenden von AWS CloudFormation zum Erstellen von Richtlinien und Rollen in IAM

Sie können Vorlagen erstellen und verwenden, AWS CloudFormation um die Richtlinien und Rollen zu erstellen, die Sie für den Zugriff auf Ressourcen in einem AWS-Konto für Ihre CodeCatalyst Projekte und Workflows benötigen. AWS CloudFormation ist ein Service, der Ihnen hilft, Ihre AWS Ressourcen zu modellieren und einzurichten, sodass Sie weniger Zeit für die Verwaltung dieser Ressourcen aufwenden müssen und sich mehr auf Ihre Anwendungen konzentrieren können, die auf ausgeführt werdenAWS. Wenn Sie beabsichtigen, Rollen in mehreren zu erstellenAWS-Konten, kann das Erstellen einer Vorlage Ihnen helfen, diese Aufgabe schneller auszuführen.

Die folgende Beispielvorlage erstellt eine Bereitstellungsaktionsrolle und -richtlinie.

```
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: '*'
```

## Manuelles Erstellen der Rolle für den Blueprint der Webanwendung

Die Vorlage für CodeCatalyst Webanwendungen verwendet IAM-Rollen, die Sie erstellen und die als Build-Rolle für CDK, die Bereitstellungsrolle und die Stack-Rolle bezeichnet werden.

Gehen Sie wie folgt vor, um die Rolle in IAM zu erstellen.

So erstellen Sie eine Build-Rolle

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich bei AWS an.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
  - c. Wählen Sie im Navigationsbereich Policies aus.
  - d. Wählen Sie Richtlinie erstellen aus.
  - e. Wählen Sie den Tab JSON.

- f. Löschen Sie den vorhandenen Code.
- g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie unter Name Folgendes ein:

```
codecatalyst-webapp-build-policy
```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Build-Rolle wie folgt:

- a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie aus.
- c. Löschen Sie die vorhandene benutzerdefinierte Vertrauensrichtlinie.
- d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Wählen Sie Weiter aus.
- f. Fügen Sie die Berechtigungsrichtlinie an die Build-Rolle an. Suchen Sie auf der Seite Berechtigungen hinzufügen im Abschnitt Berechtigungsrichtlinien nach dem entsprechenden Kontrollkästchen `codecatalyst-webapp-build-policy` und aktivieren Sie es.
- g. Wählen Sie Weiter aus.
- h. Geben Sie für Rollenname Folgendes ein:

**codecatalyst-webapp-build-role**

- i. Geben Sie für Rollenbeschreibung Folgendes ein:

**CodeCatalyst Web app build role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Fügen Sie die Berechtigungsrichtlinie wie folgt an die Build-Rolle an:
  - a. Wählen Sie im Navigationsbereich Rollen und suchen Sie dann nach `codecatalyst-webapp-build-role`.
  - b. Wählen Sie `codecatalyst-webapp-build-role`, um die Details anzuzeigen.
  - c. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen und dann Richtlinien anfügen aus.
  - d. Suchen Sie nach `codecatalyst-webapp-build-policy`, aktivieren Sie das entsprechende Kontrollkästchen und wählen Sie dann Richtlinien anfügen aus.

Sie haben nun die Berechtigungsrichtlinie an die Build-Rolle angehängt. Die Build-Rolle verfügt jetzt über zwei Richtlinien: eine Berechtigungsrichtlinie und eine Vertrauensrichtlinie.

4. Rufen Sie den ARN der Build-Rolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-webapp-build-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.

Die Seite Zusammenfassung der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.

Sie haben jetzt die Build-Rolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

## Manuelles Erstellen von Rollen für den SAM-Blueprint

Der CodeCatalyst SAM-Blueprint verwendet IAM-Rollen, die Sie erstellen und die als Build-Rolle für CloudFormation und als Bereitstellungsrolle für SAM bezeichnet werden.

Gehen Sie wie folgt vor, um die Rollen in IAM zu erstellen.

## So erstellen Sie eine Build-Rolle für CloudFormation

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:
  - a. Melden Sie sich bei AWS an.
  - b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
  - c. Wählen Sie im Navigationsbereich Policies aus.
  - d. Wählen Sie Richtlinie erstellen aus.
  - e. Wählen Sie den Tab JSON.
  - f. Löschen Sie den vorhandenen Code.
  - g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:*",
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.

- j. Geben Sie unter Name Folgendes ein:

```
codecatalyst-SAM-build-policy
```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

2. Erstellen Sie die Build-Rolle wie folgt:

- Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- Wählen Sie Benutzerdefinierte Vertrauensrichtlinie aus.
- Löschen Sie die vorhandene benutzerdefinierte Vertrauensrichtlinie.
- Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Wählen Sie Weiter aus.
- Fügen Sie die Berechtigungsrichtlinie an die Build-Rolle an. Suchen Sie auf der Seite Berechtigungen hinzufügen im Abschnitt Berechtigungsrichtlinien nach `codecatalyst-SAM-build-policy` und aktivieren Sie das entsprechende Kontrollkästchen.
- Wählen Sie Weiter aus.
- Geben Sie für Rollenname Folgendes ein:

```
codecatalyst-SAM-build-role
```



- i. Geben Sie für Rollenbeschreibung Folgendes ein:

`CodeCatalyst SAM build role`

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Fügen Sie die Berechtigungsrichtlinie wie folgt an die Build-Rolle an:
  - a. Wählen Sie im Navigationsbereich Rollen und suchen Sie dann nach `codecatalyst-SAM-build-role`.
  - b. Wählen Sie `codecatalyst-SAM-build-role`, um die Details anzuzeigen.
  - c. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen und dann Richtlinien anfügen aus.
  - d. Suchen Sie nach `codecatalyst-SAM-build-policy`, aktivieren Sie das entsprechende Kontrollkästchen und wählen Sie dann Richtlinien anfügen aus.

Sie haben nun die Berechtigungsrichtlinie an die Build-Rolle angehängt. Die Build-Rolle verfügt jetzt über zwei Richtlinien: eine Berechtigungsrichtlinie und eine Vertrauensrichtlinie.

4. Rufen Sie den ARN der Build-Rolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-SAM-build-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.

Die Seite Zusammenfassung der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.

Sie haben jetzt die Build-Rolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

So erstellen Sie eine Bereitstellungsrolle für SAM

1. Erstellen Sie wie folgt eine Richtlinie für die Rolle:

- a. Melden Sie sich bei AWS an.
- b. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- c. Wählen Sie im Navigationsbereich Policies aus.
- d. Wählen Sie Richtlinie erstellen aus.
- e. Wählen Sie den Tab JSON.
- f. Löschen Sie den vorhandenen Code.
- g. Fügen Sie folgenden Code ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

Wenn die Rolle zum ersten Mal zum Ausführen von Workflow-Aktionen verwendet wird, verwenden Sie den Platzhalter in der Ressourcenrichtlinienanweisung und schränken Sie dann die Richtlinie nach der Verfügbarkeit mit dem Ressourcennamen ein.

```
"Resource": "*"

```

- h. Wählen Sie Next: Tags (Weiter: Tags) aus.
- i. Klicken Sie auf Weiter: Prüfen.
- j. Geben Sie unter Name Folgendes ein:

```
codecatalyst-SAM-deploy-policy

```

- k. Wählen Sie Richtlinie erstellen aus.

Sie haben jetzt eine Berechtigungsrichtlinie erstellt.

## 2. Erstellen Sie die Build-Rolle wie folgt:

- a. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
- b. Wählen Sie Benutzerdefinierte Vertrauensrichtlinie aus.
- c. Löschen Sie die vorhandene benutzerdefinierte Vertrauensrichtlinie.
- d. Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- e. Wählen Sie Weiter aus.

- f. Fügen Sie die Berechtigungsrichtlinie an die Build-Rolle an. Suchen Sie auf der Seite Berechtigungen hinzufügen im Abschnitt Berechtigungsrichtlinien nach `codecatalyst-SAM-deploy-policy` und aktivieren Sie das entsprechende Kontrollkästchen.
- g. Wählen Sie Weiter aus.
- h. Geben Sie für Rollenname Folgendes ein:

**`codecatalyst-SAM-deploy-role`**

- i. Geben Sie für Rollenbeschreibung Folgendes ein:

**CodeCatalyst SAM deploy role**

- j. Wählen Sie Rolle erstellen aus.

Sie haben jetzt eine Build-Rolle mit einer Vertrauensrichtlinie und einer Berechtigungsrichtlinie erstellt.

3. Fügen Sie die Berechtigungsrichtlinie wie folgt an die Build-Rolle an:
  - a. Wählen Sie im Navigationsbereich Rollen und suchen Sie dann nach `codecatalyst-SAM-deploy-role`.
  - b. Wählen Sie `codecatalyst-SAM-deploy-role`, um die Details anzuzeigen.
  - c. Wählen Sie auf der Registerkarte Berechtigungen die Option Berechtigungen hinzufügen und dann Richtlinien anfügen aus.
  - d. Suchen Sie nach `codecatalyst-SAM-deploy-policy`, aktivieren Sie das entsprechende Kontrollkästchen und wählen Sie dann Richtlinien anfügen aus.

Sie haben nun die Berechtigungsrichtlinie an die Build-Rolle angehängt. Die Build-Rolle verfügt jetzt über zwei Richtlinien: eine Berechtigungsrichtlinie und eine Vertrauensrichtlinie.

4. Rufen Sie den ARN der Build-Rolle wie folgt ab:
  - a. Wählen Sie im Navigationsbereich Rollen aus.
  - b. Geben Sie im Suchfeld den Namen der Rolle ein, die Sie gerade erstellt haben (`codecatalyst-SAM-deploy-role`).
  - c. Wählen Sie die Rolle aus der Liste aus.

Die Seite Zusammenfassung der Rolle wird angezeigt.

- d. Kopieren Sie oben den ARN-Wert.

Sie haben jetzt die Build-Rolle mit den entsprechenden Berechtigungen erstellt und ihren ARN abgerufen.

## Konformitätsvalidierung für Amazon CodeCatalyst

Informationen darüber, ob ein AWS-Service in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter [AWS-Services in Geltungsbereich nach Compliance-Programm](#). Wählen Sie das Compliance-Programm, das Sie interessiert. Allgemeine Informationen finden Sie unter [AWS-Compliance-Programme](#).

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter [Berichte herunterladen in AWS Artifact](#).

Ihre Compliance-Verantwortung bei der Verwendung von AWS-Services ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- [Kurzanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden werden Überlegungen zur Architektur erörtert und Schritte zum Bereitstellen von Basisumgebungen auf AWS zur Verfügung gestellt, die auf Sicherheit und Compliance ausgerichtet sind.
- [Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance in Amazon Web Services](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-berechtigte Anwendungen erstellen können.

### Note

Nicht alle AWS-Services sind HIPAA-berechtigt. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS-Compliance-Ressourcen](#) – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National

Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.

- [Auswertung von Ressourcen mit Regeln](#) im AWS ConfigEntwicklerhandbuch – Der AWS Config-Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.
- [AWS Security Hub](#) – Dieser AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb von AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS-Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [AWS Audit Manager](#) – Dieser AWS-Service hilft Ihnen, Ihre AWS-Nutzung kontinuierlich zu überprüfen, um den Umgang mit Risiken und die Compliance von Branchenstandards zu vereinfachen.

## Resilienz bei Amazon CodeCatalyst

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und -Availability Zones. Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#). Weitere Informationen darüber, welche CodeCatalyst Daten repliziert werden AWS-Regionen, finden Sie unter [Datenschutz bei Amazon CodeCatalyst](#).

## Infrastruktursicherheit bei Amazon CodeCatalyst

Als verwalteter Service CodeCatalyst ist Amazon durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS-Sicherheitsservices und wie AWS die Infrastruktur schützt, finden Sie unter [AWS-Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) im Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff CodeCatalyst über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

## Konfiguration und Schwachstellenanalyse in Amazon CodeCatalyst

Konfiguration und IT-Steuerelemente unterliegen der übergreifenden Verantwortlichkeit von AWS und Ihnen, unserem Kunden. Weitere Informationen finden Sie unter [AWS Modell der übergreifenden Verantwortlichkeit](#).

## Ihre Daten und Ihr Datenschutz bei Amazon CodeCatalyst

Amazon CodeCatalyst nimmt Ihre Privatsphäre ernst und die Sicherheit Ihrer Daten hat für uns oberste Priorität. Weitere Informationen darüber, wie wir mit Ihren Daten umgehen, finden Sie in der [AWS Datenschutzerklärung](#).

Informationen zum Anfordern und Einsehen Ihrer Daten finden [Sie unter Daten anfordern](#) imAllgemeine AWS-Referenz.

## Löschen Sie Ihr AWS Builder ID-Profil

Das Löschen Ihres Profils ist eine permanente Aktion, die nicht rückgängig gemacht werden kann. Der Löschvorgang beginnt sofort, nachdem Sie Löschen ausgewählt haben. Amazon CodeCatalyst beginnt mit dem Löschen Ihres Profils und aller zugehörigen persönlichen Daten. Es kann bis zu 90 Tage dauern, bis dieser Vorgang abgeschlossen ist.

Wenn Ihr Profil gelöscht wird, können Sie nicht auf Ihre Daten in Amazon zugreifen oder sie wiederherstellen CodeCatalyst. Dazu gehören persönliche Zugriffstoken, Rollen, Benutzermitgliedschaften und alle CodeCatalyst Amazon-Bereiche, bei denen Sie das einzige Mitglied sind. Sie können sich nicht mehr bei Amazon anmelden CodeCatalyst.

Informationen zum Löschen Ihres AWS Builder-ID-Profiles finden Sie unter [Löschen Ihrer AWS Builder-ID](#) in der Allgemeinen AWS-Referenz.

## Bewährte Methoden für Workflow-Aktionen in Amazon CodeCatalyst

Es gibt eine Reihe von bewährten Sicherheitsmethoden, die Sie bei der Entwicklung Ihrer Workflows berücksichtigen sollten CodeCatalyst. Die folgenden Richtlinien sind allgemeine Richtlinien und stellen keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

### Themen

- [Vertrauliche Informationen](#)
- [Lizenzbedingungen](#)
- [Nicht vertrauenswürdiger Code](#)
- [GitHub Aktionen](#)

### Vertrauliche Informationen

Betten Sie keine vertraulichen Informationen in Ihr YAML ein. Anstatt Anmeldeinformationen, Schlüssel oder Token in Ihr YAML einzubetten, empfehlen wir Ihnen, Geheimnisse zu verwenden. CodeCatalyst Secrets bieten eine einfache Möglichkeit, vertrauliche Informationen in Ihrer YAML zu speichern und darauf zu verweisen.

### Lizenzbedingungen

Achten Sie unbedingt auf die Lizenzbedingungen der Aktion, die Sie verwenden möchten.

### Nicht vertrauenswürdiger Code

Aktionen sind in der Regel eigenständige Module mit einem einzigen Zweck, die von einem Projekt, einem Bereich oder einer breiteren Community gemeinsam genutzt werden können. Die Verwendung von Code von anderen kann ein großer Komfort- und Effizienzgewinn sein, führt aber auch zu einem neuen Bedrohungsvektor. Lesen Sie sich die folgenden Abschnitte durch, um sicherzustellen, dass Sie die bewährten Methoden befolgen, um Ihre CI/CD-Workflows zu schützen.



## GitHub Aktionen

GitHub Die Aktionen sind Open Source und werden von der Community erstellt und verwaltet. Wir folgen dem [Modell der gemeinsamen Verantwortung](#) und betrachten den Quellcode von GitHub Actions als Kundendaten, für die Sie verantwortlich sind. GitHub Aktionen können Zugriff auf Geheimnisse, Repository-Token, Quellcode, Kontolinks und Ihre Rechenzeit gewährt werden. Stellen Sie sicher, dass Sie von der Vertrauenswürdigkeit und Sicherheit der GitHub Aktionen, die Sie ausführen möchten, überzeugt sind.

Spezifischere Leitlinien und bewährte Sicherheitsverfahren für GitHub Aktionen:

- [Verschärfung der Sicherheit](#)
- [Verhinderung eigener Anfragen](#)
- [Nicht vertrauenswürdige Eingaben](#)
- [Wie können Sie Ihren Bausteinen vertrauen](#)

## CodeCatalyst Vertrauensmodell

Das CodeCatalyst Vertrauensmodell von Amazon CodeCatalyst ermöglicht es, die Servicerolle im verbundenen Bereich zu übernehmenAWS-Konto. Das Modell verbindet die IAM-Rolle, die CodeCatalyst Serviceprinzipale und den CodeCatalyst Bereich. Die Vertrauensrichtlinie verwendet den `aws:SourceArn` Bedingungsschlüssel, um Berechtigungen für den im Bedingungsschlüssel angegebenen CodeCatalyst Bereich zu gewähren. Weitere Informationen zu diesem Bedingungsschlüssel finden Sie unter [aws: SourceArn](#) im IAM-Benutzerhandbuch.

Eine Vertrauensrichtlinie ist ein JSON-Richtliniendokument, in dem Sie die Prinzipale definieren, denen Sie vertrauen, dass sie die Rolle übernehmen. Eine Rollenvertrauensrichtlinie ist eine erforderliche ressourcenbasierte Richtlinie die einer Rolle in IAM angefügt ist. Weitere Informationen finden Sie unter [Begriffe und Konzepte](#) im IAM-Benutzerhandbuch. Einzelheiten zu den Service Principals für finden Sie CodeCatalyst unter. [Dienstprinzipale für CodeCatalyst](#)

In der folgenden Vertrauensrichtlinie werden den im `Principal` Element aufgelisteten Dienstprinzipalen Berechtigungen aus der ressourcenbasierten Richtlinie gewährt, und der `Condition` Block wird verwendet, um den Zugriff auf die Ressource mit eingeschränktem Gültigkeitsbereich zu beschränken.

```
"Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": "iam:PassRole",  
      "Resource": "arn:aws:iam::123456789012:role/CodeCatalystRole",  
      "Condition": {  
        "StringEquals": {  
          "aws:SourceArn": "arn:aws:codecatalyst:us-east-1:123456789012:service-principal/CodeCatalystServicePrincipal" }  
        }  
      }  
    ]
```

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

In der Vertrauensrichtlinie erhalten die CodeCatalyst Service Principals Zugriff über den `aws:SourceArn` Bedingungsschlüssel, der den Amazon-Ressourcennamen (ARN) für die CodeCatalyst Space-ID enthält. Der ARN verwendet das folgende Format:

```
arn:aws:codecatalyst:::space/spaceId/project/*
```

#### Important

Verwenden Sie die Space-ID nur in Bedingungsschlüsseln, wie `aws:SourceArn` z.  
Verwenden Sie die Space-ID in IAM-Richtlinienanweisungen nicht als Ressourcen-ARN.

Es hat sich bewährt, die Berechtigungen in der Richtlinie so weit wie möglich einzuschränken.

- Sie können den Platzhalter (\*) im `aws:SourceArn` Bedingungsschlüssel verwenden, um alle Projekte in dem Bereich mit `project/*` anzugeben.
- Sie können im `aws:SourceArn` Bedingungsschlüssel Berechtigungen auf Ressourcenebene für ein bestimmtes Projekt im Bereich mit angeben. `project/projectId`

## Dienstprinzipale für CodeCatalyst

Sie verwenden das `Principal` Element in einer ressourcenbasierten JSON-Richtlinie, um den Prinzipal anzugeben, dem der Zugriff auf eine Ressource gewährt oder verweigert wird. Zu den

Auftraggeber, die Sie in der Vertrauensrichtlinie angeben können, gehören Benutzer, Rollen, Konten und Services. Sie können das `Principal` Element nicht in einer identitätsbasierten Richtlinie verwenden. Ebenso können Sie eine Benutzergruppe nicht als Principal in einer Richtlinie (z. B. einer ressourcenbasierten Richtlinie) identifizieren, da sich Gruppen auf Berechtigungen und nicht auf Authentifizierung beziehen und Prinzipale authentifizierte IAM-Entitäten sind.

In der Vertrauensrichtlinie können Sie im `Principal` Element einer ressourcenbasierten Richtlinie AWS-Services oder in Bedingungsschlüsseln angeben, dass Prinzipale unterstützt werden. Dienstprinzipale werden durch den Dienst definiert. Im Folgenden sind die Dienstprinzipale definiert für: CodeCatalyst

- `codecatalyst.amazonaws.com` — Dieser Service Principal wird für eine Rolle verwendet, die Zugriff gewährt auf. CodeCatalyst AWS
- `codecatalyst-runner.amazonaws.com` — Dieser Service Principal wird für eine Rolle verwendet, die Zugriff auf Ressourcen in Bereitstellungen für Workflows gewährt. CodeCatalyst AWS  
CodeCatalyst

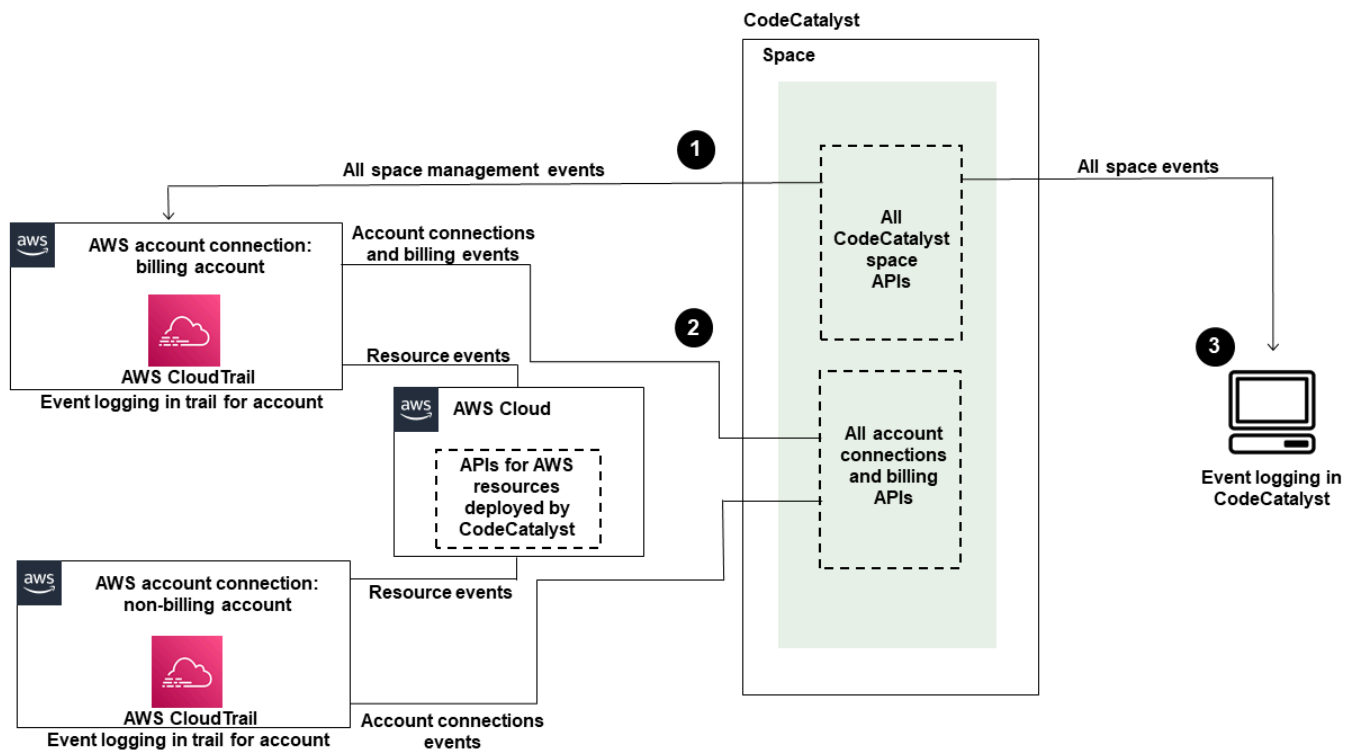
Weitere Informationen finden Sie unter [AWSJSON-Richtlinienelemente: Principal im IAM-Benutzerhandbuch](#).

## Überwachung bei Amazon CodeCatalyst

In Amazon CodeCatalyst werden Verwaltungsereignisse für den Space vom Rechnungskonto für den Space gesammelt AWS CloudTrail und dort protokolliert. CloudTrail Die Protokollierung ist die primäre Methode zur Verwaltung der Protokollierung von CodeCatalyst Ereignissen, und eine sekundäre Methode ist die Anzeige der Ereignisprotokollierung CodeCatalyst.

Ereignisse im Konto werden mit dem Trail und dem dafür vorgesehenen Bucket protokolliert AWS-Konto.

Das folgende Diagramm zeigt, wie alle Verwaltungsereignisse für den Bereich CloudTrail für das Abrechnungskonto protokolliert werden, während Kontoverbindungen/Abrechnungsereignisse und AWS Ressourcenereignisse CloudTrail für das jeweilige Konto protokolliert werden.



Die Abbildung zeigt die folgenden Schritte:

1. Wenn ein Bereich erstellt wird, AWS-Konto wird ein Bereich mit dem Bereich verbunden und als Rechnungskonto festgelegt. Bei dem verwendeten Trail handelt es sich um den Trail, der CloudTrail für das Rechnungskonto erstellt wurde und in dem Space-Ereignisse protokolliert werden. CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen eines CodeCatalyst Space getätigt wurden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen S3-Bucket. Wenn das Rechnungskonto in ein anderes AWS-Konto geändert wird, werden Space-Ereignisse im Trail und Bucket für dieses Konto protokolliert. Weitere Informationen zu CodeCatalyst Verwaltungsereignissen, die von protokolliert werden CloudTrail, finden Sie unter [CodeCatalyst Informationen in CloudTrail](#).
2. Andere Konten, die mit dem Bereich verbunden sind, einschließlich des Rechnungskontos, protokollieren eine Teilmenge der Ereignisse für Kontoverbindungen und Abrechnungsergebnisse. CodeCatalyst Workflows, die Kontoereignisse für AWS Ressourcen generieren, die für dieses Konto bereitgestellt wurden, werden ebenfalls im Trail und Bucket für den AWS-Konto protokolliert. CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen eines CodeCatalyst Space getätigt wurden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen S3-Bucket. Weitere Informationen zu CodeCatalyst Verwaltungsereignissen, die

von protokolliert werden CloudTrail, finden Sie unter [Zugreifen auf protokollierte Ereignisse in CodeCatalyst](#).

3. Sie können auch CodeCatalyst Aktionen in Ihrem Bereich innerhalb eines bestimmten Zeitraums überwachen, indem Sie den [list-event-logs](#) Befehl verwenden AWS CLI. Weitere Informationen finden Sie [im Amazon CodeCatalyst API-Referenzhandbuch](#). Sie benötigen die Rolle eines Space-Administrators, um die Liste der Ereignisse für CodeCatalyst Aktionen in Ihrem Bereich aufrufen zu können. Weitere Informationen finden Sie unter [Zugreifen auf protokollierte Ereignisse in CodeCatalyst](#).

#### Note

ListEventLogsgarantiert Ereignisse der letzten 30 Tage in einem bestimmten Bereich. Sie können auch eine Liste der Verwaltungsereignisse der letzten 90 Tage CodeCatalyst in der AWS CloudTrail Konsole anzeigen und abrufen, indem Sie den Ereignisverlauf aufrufen oder einen Pfad erstellen, um Ereignisse der letzten 90 Tage aufzuzeichnen und zu verwalten. Weitere Informationen finden Sie unter [Arbeiten mit dem CloudTrail Ereignisverlauf](#) und [Arbeiten mit CloudTrail Pfaden](#).

#### Note

AWS Ressourcen, die in verbundenen Konten für CodeCatalyst Workflows bereitgestellt werden, werden nicht im Rahmen der CloudTrail Protokollierung für den CodeCatalyst Bereich protokolliert. Zu den CodeCatalyst Ressourcen gehören beispielsweise ein Bereich oder ein Projekt. AWS Zu den Ressourcen gehören ein Amazon ECS-Service oder eine Lambda-Funktion. Sie müssen die CloudTrail Protokollierung für jeden Standort, in AWS-Konto dem Ressourcen bereitgestellt werden, separat konfigurieren.

Hier ist ein möglicher Ablauf für die Ereignisüberwachung in CodeCatalyst.

Mary Major ist Bereichsadministratorin für einen CodeCatalyst Bereich und sieht sich alle Verwaltungsereignisse CodeCatalyst für Ressourcen auf Bereichs- und Projektebene in dem Bereich an, die angemeldet sind. CloudTrail Sehen Sie sich [CodeCatalyst Informationen in CloudTrail](#) zum Beispiel Ereignisse an, die angemeldet sind. CloudTrail

Bei Ressourcen, die in erstellt wurden CodeCatalyst, wie z. B. Entwicklungsumgebungen, sieht sich Mary den Ereignisverlauf im Abrechnungskonto für den Bereich an und untersucht Ereignisse, in

CodeCatalyst den Entwicklungsumgebungen von Projektmitgliedern erstellt wurden. Das Ereignis stellt den Identitätsspeicher, den IAM-Identitätstyp und die Anmeldeinformationen für die AWS Builder-ID für den Benutzer bereit, der die Entwicklungsumgebung erstellt hat. Für Ressourcen, die AWS bei der Bereitstellung durch Workflows in erstellt werden CodeCatalyst, wie z. B. eine Lambda-Funktion für eine serverlose Bereitstellung, kann der AWS-Konto Besitzer den Ereignisverlauf für den Trail einsehen, der mit der separaten Aktion AWS-Konto (die auch ein verbundenes Konto ist CodeCatalyst) für die Workflow-Bereitstellungsaktion verknüpft ist.

Um weitere Informationen zu erhalten, kann Mary mithilfe des [list-event-logs](#)Befehls im auch Ereignisse für alle CodeCatalyst APIs im Space anzeigen. AWS CLI

Themen

- [Protokollieren von CodeCatalyst API-Aufrufen in Connected AWS-Konten mit AWS CloudTrail](#)
- [Zugreifen auf protokollierte Ereignisse in CodeCatalyst](#)

## Protokollieren von CodeCatalyst API-Aufrufen in Connected AWS-Konten mit AWS CloudTrail

Amazon CodeCatalyst ist in einen Service integriertAWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem ausgeführten Aktionen bereitstelltAWS-Service. CloudTrail erfasst API-Aufrufe, die im Namen von CodeCatalyst in Connected getätigt wurdenAWS-Konten, als Ereignisse. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen S3-Bucket aktivieren, einschließlich Ereignissen für CodeCatalyst. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen.

CodeCatalyst unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail Protokolldateien:

- Verwaltungsereignisse für CodeCatalyst Spaces werden in dem AWS-Konto dafür vorgesehenen Abrechnungskonto protokolliert. Weitere Informationen finden Sie unter [CodeCatalyst Ereignisse im Weltraum](#).

### Note

Auf Datenereignisse für CodeCatalyst Leerzeichen kann über die CLI zugegriffen werden, wie unter beschrieben [Zugreifen auf protokollierte Ereignisse in CodeCatalyst](#).

- Ereignisse für Ressourcen, die in CodeCatalyst Workflow-Aktionen verwendet werden, die in einer Verbindung auftreten, AWS-Konto werden dort als Ereignisse protokolliert AWS-Konto. Weitere Informationen finden Sie unter [CodeCatalyst Kontoverbindungen und Abrechnungseignisse](#).

#### Important

Einem Bereich können zwar mehrere Konten zugeordnet werden, die CloudTrail Protokollierung von Ereignissen in CodeCatalyst Bereichen und Projekten gilt jedoch nur für das Rechnungskonto.

Das Space-Abrechnungskonto gehört Ihnen AWS-Konto, für das CodeCatalyst Ressourcen berechnet werden, die über das AWS kostenlose Kontingent hinausgehen. Es können mehrere Konten mit einem Space verbunden werden, wobei nur ein Konto das angegebene Rechnungskonto sein kann. Das Rechnungskonto oder weitere verbundene Konten für den Bereich können IAM-Rollen haben, die für die Bereitstellung von AWS Ressourcen und Infrastruktur, wie z. B. einem Amazon ECS-Cluster oder S3-Bucket, aus CodeCatalyst Workflows verwendet werden. Sie können das Workflow-YAML verwenden, um das zu identifizieren, für das Sie AWS-Konto die Bereitstellung vorgenommen haben.

#### Note

AWS Ressourcen, die in verbundenen Konten für CodeCatalyst Workflows bereitgestellt werden, werden nicht im Rahmen der CloudTrail Protokollierung für den CodeCatalyst Bereich protokolliert. Zu den CodeCatalyst Ressourcen gehören beispielsweise ein Bereich oder ein Projekt. AWS Zu den Ressourcen gehören ein Amazon ECS-Service oder eine Lambda-Funktion. CloudTrail Die Protokollierung muss für jeden Standort, in AWS-Konto dem Ressourcen bereitgestellt werden, separat konfiguriert werden.

CodeCatalyst Die Anmeldung bei verbundenen Konten beinhaltet die folgenden Überlegungen:

- Der Zugriff auf CloudTrail Ereignisse wird mit IAM im verbundenen Konto verwaltet und nicht in CodeCatalyst.
- Verbindungen von Drittanbietern, wie z. B. das Verknüpfen mit einem GitHub Repository, führen dazu, dass Ressourcennamen von Drittanbietern in CloudTrail Protokollen aufgezeichnet werden.

**Note**

CloudTrail Die Protokollierung von CodeCatalyst Ereignissen erfolgt auf Bereichsebene und isoliert Ereignisse nicht nach Projektgrenzen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrailBenutzerhandbuch](#).

**Note**

In diesem Abschnitt wird die CloudTrail Protokollierung aller Ereignisse beschrieben, die in einem CodeCatalyst Bereich protokolliert wurden und mit AWS-Konten denen eine Verbindung hergestellt wurde CodeCatalyst. Um alle in einem CodeCatalyst Bereich protokollierten Ereignisse zu überprüfen, können Sie außerdem den Befehl AWS CLI und den `aws codecatalyst list-event-logs` Befehl verwenden. Weitere Informationen finden Sie unter [Zugreifen auf protokollierte Ereignisse in CodeCatalyst](#).

## CodeCatalyst Ereignisse im Weltraum

Aktionen CodeCatalyst zur Verwaltung von Ressourcen auf Bereichs- und Projektebene werden im Rechnungskonto für den Bereich protokolliert. Bei der CloudTrail Protokollierung für einen CodeCatalyst Bereich werden Ereignisse unter Berücksichtigung der folgenden Überlegungen protokolliert.

- CloudTrail Ereignisse gelten für den gesamten Bereich und beziehen sich nicht auf ein einzelnes Projekt.
- Wenn Sie eine Verbindung AWS-Konto zu einem CodeCatalyst Bereich herstellen, werden protokollierbare Ereignisse für Kontoverbindungen in diesem Bereich protokolliert. AWS-Konto Nachdem Sie diese Verbindung aktiviert haben, können Sie sie nicht mehr deaktivieren.
- Wenn Sie eine Verbindung AWS-Konto zu einem CodeCatalyst Bereich herstellen und ihn als Rechnungskonto für den Bereich festlegen, werden Ereignisse in diesem AWS-Konto Bereich protokolliert. Nachdem Sie diese Verbindung aktiviert haben, können Sie sie nicht mehr deaktivieren.

Ereignisse für Ressourcen auf Bereichs- und Projektebene werden nur im Rechnungskonto protokolliert. Um das CloudTrail Zielkonto zu ändern, aktualisieren Sie das Rechnungskonto unter CodeCatalyst Zu Beginn des nächsten monatlichen Abrechnungszeitraums wird die Änderung



für das neue Rechnungskonto in wirksam CodeCatalyst. Danach wird das CloudTrail Zielkonto aktualisiert.

Im Folgenden finden Sie Beispiele für EreignisseAWS, die sich auf Aktionen CodeCatalyst zur Verwaltung von Ressourcen auf Raum- und Projektebene beziehen. Die folgenden APIs werden über das SDK und die CLI veröffentlicht. Ereignisse werden in dem als Rechnungskonto für den CodeCatalyst Bereich AWS-Konto angegebenen Konto protokolliert.

- [CreateDevEnvironment](#)
- [CreateProject](#)
- [DeleteDevEnvironment](#)
- [GetDevEnvironment](#)
- [GetProject](#)
- [GetSpace](#)
- [GetSubscription](#)
- [ListDevEnvironments](#)
- [ListDevEnvironmentSessions](#)
- [ListEventLogs](#)
- [ListProjects](#)
- [ListSourceRepositories](#)
- [StartDevEnvironment](#)
- [StartDevEnvironmentSession](#)
- [StopDevEnvironment](#)
- [StopDevEnvironmentSession](#)
- [UpdateDevEnvironment](#)

## CodeCatalyst Kontoverbindungen und Abrechnungseignisse

Im Folgenden finden Sie Beispiele für EreignisseAWS, die sich auf Aktionen in CodeCatalyst Bezug auf Kontoverbindungen oder Rechnungsstellung beziehen:

- `AcceptConnection`
- `AssociateIAMRoletoConnection`

- DeleteConnection
- DissociateIAMRolefromConnection
- GetBillingAuthorization
- GetConnection
- GetPendingConnection
- ListConnections
- ListIAMRolesforConnection
- PutBillingAuthorization
- RejectConnection

## CodeCatalyst Informationen in CloudTrail

CloudTrail ist aktiviert, AWS-Konto wenn Sie dieses Konto erstellen. Wenn Sie das AWS-Konto mit einem CodeCatalyst Bereich verbinden, AWS-Konto werden Ereignisse für diesen Bereich, die in diesem Bereich auftreten, in CloudTrail diesem AWS-Konto protokolliert. Protokollierbare Ereignisse CodeCatalyst werden zusammen mit anderen CloudTrail protokollierbaren CloudTrail Ereignissen in diesem Konto als Ereignisse in den AWS Protokollen des verbundenen Kontos und im Ereignisverlauf in der CloudTrail Konsole aufgezeichnet.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage von einem Benutzer mit seiner Builder-ID gestellt wurde. AWS
- Ob die Anfrage mit Stammbenutzer- oder AWS Identity and Access Management (IAM)-Anmeldeinformationen ausgeführt wurde.
- Ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer ausgeführt wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter [CloudTrail -Element userIdentity](#).

## Auf CloudTrail Ereignisse zugreifen

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, einschließlich der Ereignisse für CodeCatalyst Aktivitäten in dem AWS-Konto, erstellen Sie einen Pfad. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien in einem S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien für den von Ihnen angegebenen S3 Bucket bereit. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Ein Trail ist eine Konfiguration, die die Übermittlung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

### Beispiel für CodeCatalyst ein Kontoverbindungsereignis in AWS

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `ListConnections` Aktion demonstriert. Für ein AWS-Konto, das mit dem Space verbunden ist, wird `ListConnections` verwendet, um alle Kontoverbindungen CodeCatalyst zu diesem Bereich anzuzeigen. Das Ereignis wird wie ein AWS-Konto angegeben, `accountId`, und der Wert von `arn` ist der Amazon-Ressourcenname (ARN) der Rolle, die für die Aktion verwendet wurde.

```
{
  "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"
}

```

## Beispiel für ein CodeCatalyst Projektressourcen-Ereignis in AWS

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateDevEnvironment` Aktion demonstriert. Ein KontoAWS-Konto, das mit dem Space verbunden ist und das angegebene Rechnungskonto für den Space ist, wird für Ereignisse auf Projektebene in dem Space verwendet, z. B. für die Erstellung einer Entwicklungsumgebung.

`userIdentity`, im `accountId` Feld, ist dies die IAM Identity Center-Konto-ID (432677196278), die den Identitätspool für alle AWS Builder-ID-Identitäten hostet. Diese Konto-ID enthält die folgenden Informationen über den CodeCatalyst Benutzer für das Ereignis.

- Das `type` Feld gibt den Typ der IAM-Entität für die Anfrage an. Für CodeCatalyst Ereignisse für Raum- und Projektressourcen ist `IdentityCenterUser` dieser Wert. Das `accountId` Feld gibt das Konto an, dem die Entität gehört, die zum Abrufen der Anmeldeinformationen verwendet wurde.
- Das `userId` Feld enthält die AWS Builder-ID-ID für den Benutzer.
- Das `identityStoreArn` Feld enthält den Rollen-ARN für das Identitätsspeicher-Konto und den Benutzer.

Das `recipientAccountId` Feld enthält die Konto-ID für das Rechnungskonto für den Space, mit einem Beispielwert hier von 111122223333.

Weitere Informationen finden Sie unter [CloudTrail -Element `userIdentity`](#).

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "IdentityCenterUser",
    "accountId": "432677196278",
    "onBehalfOf": {
      "userId": "user-ID",
      "identityStoreArn": "arn:aws:identitystore::432677196278:identitystore/d-9067642ac7"
    },
    "credentialId": "ABCDefGhiJKLMn11Lmn_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 bestimmten Fällen ist der Benutzeragent möglicherweise nicht bekannt. In diesem Fall CodeCatalyst wird Unknown im userAgent Feld des CloudTrail Ereignisses der Wert von angegeben.


## Deine CodeCatalyst Event-Trails werden abgefragt

Sie können Abfragen für Ihre CloudTrail Protokolle mithilfe einer Abfragetabelle in Amazon Athena erstellen und verwalten. Weitere Informationen zum Erstellen einer Abfrage finden Sie unter [Abfragen von AWS CloudTrail Protokollen](#) im Amazon Athena Athena-Benutzerhandbuch.

## Zugreifen auf protokollierte Ereignisse in CodeCatalyst


Wenn Benutzer Aktionen in Amazon ausführen CodeCatalyst, werden diese Aktionen als Ereignisse aufgezeichnet. Sie können den verwenden AWS CLI , um Protokolle von Ereignissen in einem bestimmten Bereich in einem bestimmten Zeitraum anzuzeigen. Sie können diese Ereignisse anzeigen, um die in dem Bereich durchgeführten Aktionen zu überprüfen, einschließlich Datum und

Uhrzeit der Aktionen, des Namens des Benutzers, der die Aktion ausgeführt hat, und der IP-Adresse, von der der Benutzer die Anfrage gestellt hat.

 Note

Verwaltungsereignisse für einen CodeCatalyst Bereich werden CloudTrail für das verbundene Rechnungskonto protokolliert. Weitere Informationen zu CodeCatalyst Verwaltungsereignissen, die von protokolliert werden CloudTrail, finden Sie unter [CodeCatalyst Informationen in CloudTrail](#).

Um ein Ereignisprotokoll für einen Bereich anzeigen zu können, müssen Sie den installiert und AWS CLI mit einem Profil für konfiguriert haben CodeCatalyst, und Sie müssen über die Rolle des Space-Administrators für den Bereich verfügen. Weitere Informationen finden Sie unter [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#) und [Rolle „Space-Administrator“](#).

 Note

Um die Protokollierung von Ereignissen anzuzeigen, die im Namen von CodeCatalyst in connected auftreten AWS-Konten, oder um die Protokollierung von Ereignissen für Space- oder Projektressourcen im verbundenen Rechnungskonto anzuzeigen, können Sie verwenden AWS CloudTrail. Weitere Informationen finden Sie unter [Protokollieren von CodeCatalyst API-Aufrufen in Connected AWS-Konten mit AWS CloudTrail](#).

1. Öffnen Sie ein Terminal oder eine Befehlszeile und führen Sie den `aws codecatalyst list-event-logs` Befehl aus. Geben Sie dabei Folgendes an:
  - Der Name des Bereichs mit der **--space-name** Option.
  - Das Datum und die Uhrzeit, zu der Sie mit der Überprüfung von Ereignissen beginnen möchten, im UTC-Zeitstempelformat (Coordinated Universal Time), wie in [RFC 3339](#) angegeben, mit der Option. **--start-time**
  - Das Datum und die Uhrzeit, zu der Sie die Überprüfung von Ereignissen beenden möchten, im UTC-Zeitstempelformat (Coordinated Universal Time), wie in [RFC 3339](#) angegeben, mit der Option. **--end-time**
  - (Optional) Die maximale Anzahl von Ergebnissen, die in einer einzigen Antwort zurückgegeben werden sollen, mit der Option. **--max-results** Wenn die Anzahl der Ergebnisse größer als

die von Ihnen angegebene Zahl ist, enthält die Antwort ein `nextToken` Element, mit dem Sie die nächsten Ergebnisse zurückgeben können.

- (Optional) Beschränken Sie die Ergebnisse mit der `--event-name` Option auf einen bestimmten Ereignistyp, den Sie zurückgeben möchten.

In diesem Beispiel werden protokollierte Ereignisse im angegebenen Bereich *ExampleCorp* aus dem Zeitraum *30.11.2022* bis *01.012.2022* zurückgegeben, wobei in der Antwort maximal *2* Ereignisse zurückgegeben werden.

```
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. Wenn in diesem Zeitraum Ereignisse aufgetreten sind, gibt der Befehl Ergebnisse zurück, die den folgenden ähneln:

```
{
  "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-EXAMPLEeaaaa",
      "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. Führen Sie den `list-event-logs` Befehl erneut mit der `--next-token` Option und dem Wert des zurückgegebenen Tokens aus, um die nächsten protokollierten Ereignisse abzurufen, die der Anforderung entsprechen.

## Kontingente für Identität, Erlaubnis und Zugriff in CodeCatalyst

In der folgenden Tabelle werden Kontingente und Beschränkungen für Identität, Berechtigungen und Zugriff in Amazon beschrieben CodeCatalyst. Weitere Informationen zu Kontingenten bei Amazon CodeCatalyst finden Sie unter [Kontingente für CodeCatalyst](#).

| Ressource   | Informationen   |
|---|---|
| Aliase in CodeCatalyst  | <p>Jede Kombination zulässiger Zeichen mit einer Länge zwischen 3 und 100 Zeichen, die mit einem Buchstaben beginnen muss. Gültige Zeichen: A-Z, a-z und 0-9. Aliase können nicht:</p> <ul style="list-style-type: none"> <li>weniger als 3 Zeichen enthalten</li> <li>Leerzeichen oder eines der folgenden Zeichen enthalten: ? ^ * [ \ ~ :</li> </ul> |
| Maximale Anzahl von Einladungen, die von einem Benutzer pro Tag gesendet werden     | 500   |
| Maximale Anzahl von Einladungen, die pro Tag an eine E-Mail-Adresse gesendet werden | 25  |
| Maximale Anzahl von Personal Access Tokens (PAT) pro Benutzer                       | 100   |
| Passwörter in CodeCatalyst  | <p>Jede Kombination zulässiger Zeichen mit einer Länge zwischen 8 und 64 Zeichen. Gültige Zeichen: A-Z, a-z und 0-9. Ihr Passwort kann die folgenden nicht alphanumerischen Zeichen enthalten: ( ~ ! @ # \$ % ^ &amp; * _ - + = `   \ { } [ ] : ; " ' &lt; &gt; , . ? / )</p>   |
| PAT-Namen in CodeCatalyst   | Beliebige Kombination von zulässigen Zeichen zwischen 1 und 100 Zeichen   |
| Zeit, bis die Einladung eines Projektmitglieds abläuft                              | Läuft nach 24 Stunden ab  |
| Zeit, bis die Einladung eines Space-Mitglieds abläuft                               | Läuft nach 24 Stunden ab  |
| Zeit, bis die Überprüfung einer E-Mail-Adresse abläuft                              | Läuft 10 Minuten nach dem Senden ab   |

# Fehlerbehebung

In diesem Abschnitt können Sie einige häufig auftretende Probleme beheben, die beim Zugriff auf Ihr CodeCatalyst Amazon-Profil auftreten können.

## Probleme bei der Anmeldung

Möglicherweise treten bei der Anmeldung einige Probleme auf. Wir haben einige Lösungen.

### Meine E-Mail-Adresse wird bereits verwendet

Wenn die von Ihnen eingegebene E-Mail bereits verwendet wird und Sie sie als Ihre eigene erkennen, haben Sie möglicherweise bereits ein Profil bei uns. Melden Sie sich mit dieser vorhandenen Identität an. Wenn Sie die bestehende E-Mail-Adresse nicht besitzen, melden Sie sich mit einer anderen, unbenutzten an.

### Ich kann die E-Mail-Verifizierung nicht abschließen

Wenn du deine Bestätigungs-E-Mail nicht erhalten hast

1. Prüfen Sie Ihren Spam- und Junk-Ordner sowie den Ordner für gelöschte Objekte.

#### Note

Diese Bestätigungs-E-Mail stammt entweder von der Adresse `no-reply@signin.aws` oder `no-reply@login.awsapps.com`. Wir empfehlen Ihnen, Ihr E-Mail-System so zu konfigurieren, dass es E-Mails von diesen Absender-E-Mail-Adressen akzeptiert und sie nicht als Junk oder Spam behandelt.

2. Warten Sie 5 Minuten und aktualisieren Sie Ihren Posteingang. Überprüfen Sie erneut Ihren Ordner für Spam, Junk und gelöschte Objekte.
3. Wenn du deine Bestätigungs-E-Mail immer noch nicht siehst, wähle Code erneut senden. Wenn Sie diese Seite bereits verlassen haben, starten Sie Ihren Workflow für die Registrierung bei [Amazon CodeCatalyst](#) neu.

### Mein Passwort erfüllt nicht die Mindestanforderungen

Zu Ihrer eigenen Sicherheit muss Ihr Passwort 8 bis 20 Zeichen, sowohl Groß- als auch Kleinbuchstaben, und Zahlen enthalten.

## Probleme bei der Anmeldung

### Ich habe mein Passwort vergessen

Führen Sie die Schritte unter [Ich habe mein Passwort vergessen](#) aus.

### Mein Passwort funktioniert nicht

Sie müssen die folgenden Anforderungen erfüllen, wenn Sie Ihr Passwort einrichten oder ändern:

- Bei Passwörtern wird zwischen Groß- und Kleinschreibung unterschieden.
- Passwörter müssen zwischen 8 und 64 Zeichen lang sein und sowohl Groß- als auch Kleinbuchstaben, Zahlen und mindestens ein nicht alphanumerisches Zeichen enthalten.
- Sie können die letzten drei Passwörter nicht wiederverwenden .

### Ich kann MFA nicht aktivieren

Um MFA zu aktivieren, fügen Sie Ihrem Profil ein oder mehrere MFA-Geräte hinzu, indem Sie den Schritten in [Multi-Faktor-Authentifizierung \(MFA\) bei Amazon CodeCatalyst](#) befolgen.

### Ich kann kein MFA-Gerät hinzufügen

Wenn Sie feststellen, dass Sie kein weiteres MFA-Gerät hinzufügen können, hat es möglicherweise die maximale Anzahl von MFA-Geräten erreicht, die Sie registrieren können. Möglicherweise müssen Sie ein vorhandenes MFA-Gerät entfernen, bevor Sie ein neues hinzufügen können.

### Ich kann ein MFA-Gerät nicht entfernen

Wenn Sie MFA deaktivieren möchten, fahren Sie mit dem Entfernen Ihres MFA-Geräts fort, indem Sie die Schritte in [Löschen eines MFA-Geräts](#) befolgen. Wenn Sie jedoch MFA aktiviert lassen möchten, sollten Sie ein weiteres MFA-Gerät hinzufügen, bevor Sie versuchen, ein vorhandenes MFA-Gerät zu entfernen. Weitere Informationen zum Hinzufügen von weiteren MFA-Geräten finden Sie unter [Wie registriere ich ein Gerät für die Verwendung mit Multi-Faktor-Authentifizierung](#).

## Probleme beim Abmelden

### Ich kann nicht finden, wo ich mich abmelden kann

Wählen Sie in der oberen rechten Ecke der Seite Abmelden aus.

## Mit der Abmeldung werde ich nicht vollständig abgemeldet

Das System ist so konzipiert, dass es sich sofort abmeldet, aber die vollständige Abmeldung kann bis zu einer Stunde dauern.

## Ich erhalte die Fehlermeldung „Rolle existiert nicht“ für einen fehlgeschlagenen Workflow

Problem: Nach dem Erstellen eines Projekts aus der Webanwendung oder dem serverlosen Blueprint schlägt der Workflow mit dem folgenden Fehler fehl:

CLIENT\_ERROR: Die Rolle ist nicht vorhanden

Mögliche Lösung: Nachdem Sie eine IAM-Rolle mit den Berechtigungen zur Ausführung Ihres Workflows konfiguriert und die IAM-Rolle zu Ihrem Workflow-YAML hinzugefügt haben, schlägt der Workflow immer noch fehl, da die IAM-Rolle möglicherweise zu Ihrer Kontoverbindung hinzugefügt werden muss. Fügen Sie die IAM-Rolle zur Kontoverbindung für Ihren Bereich hinzu, wie unter beschrieben. [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#)

## Ich erhalte einen Rollenfehler für einen fehlgeschlagenen Workflow

Problem: Nach dem Erstellen eines Projekts aus der Webanwendung oder dem serverlosen Blueprint schlägt der Workflow mit dem folgenden Fehler fehl:

CLIENT\_ERROR: Die Rolle wurde nicht richtig eingerichtet oder ist nicht vorhanden

Mögliche Lösung: Für den Bereich, in dem das Projekt erstellt wurde, muss möglicherweise eine AWS-Konto Verbindung eingerichtet oder eine Kontoverbindungsanfrage abgeschlossen werden. Wenn Ihr Bereich bereits über eine aktive AWS-Konto Verbindung verfügt, erstellen und fügen Sie eine IAM-Rolle mit Berechtigungen zum Ausführen von Workflow-Aktionen hinzu. Fügen Sie die IAM-Rolle zu Ihrer Kontoverbindung hinzu, wie unter beschrieben. [Hinzufügen von IAM-Rollen zu Kontoverbindungen](#)

Mögliche Lösung: Wenn das Projekt ohne Angabe einer Verbindung erstellt wurde, muss die Kontoverbindung mit der Bereitstellungsumgebung verknüpft werden. Wenn Ihr Bereich bereits über eine aktive AWS-Konto Verbindung verfügt und eine IAM-Rolle hinzugefügt wurde, müssen Sie die Kontoverbindung mit der IAM-Rolle zu Ihrer Bereitstellungsumgebung hinzufügen, wie unter beschrieben. [Hinzufügen der Kontoverbindung und der IAM-Rollen zu Ihrer Bereitstellungsumgebung](#)

## Ich muss die IAM-Rolle in einem Projekt-Workflow aktualisieren

Wenn die AWS-Konto Verbindung vollständig eingerichtet ist und die IAM-Rolle erstellt und der Kontoverbindung hinzugefügt wurde, können Sie die IAM-Rolle in Ihrem Projekt-Workflow aktualisieren.

1. Wählen Sie die CI/CD-Option und wählen Sie Ihren Workflow. Wählen Sie die YAML-Schaltfläche.
2. Wählen Sie Bearbeiten aus.
3. Ersetzen Sie in dem `ActionRoleArn`: Feld den IAM-Rollen-ARN durch den aktualisierten IAM-Rollen-ARN. Wählen Sie Validieren aus.
4. Wählen Sie Commit (Übergeben).

Der Workflow startet automatisch, wenn er sich in der Hauptleitung befindet. Andernfalls wählen Sie Ausführen, um den Workflow erneut auszuführen.

## Wie fülle ich ein Support-Formular aus?

Sie können zu [Amazon](#) gehen CodeCatalyst oder ein [Support-Feedback-Formular](#) ausfüllen. Geben Sie im Abschnitt Informationen anfordern unter Wie können wir Ihnen helfen, an, dass Sie ein CodeCatalyst Amazon-Kunde sind. Geben Sie so viele Details wie möglich an, damit wir Ihr Problem am effizientesten lösen können.

# Erweiterungen in CodeCatalyst

Amazon CodeCatalyst bietet Erweiterungen, mit denen Sie Funktionen hinzufügen und in Produkte außerhalb von integrieren können CodeCatalyst. Mit den Erweiterungen aus dem CodeCatalyst Katalog können Teams ihre Erlebnisse individuell anpassen CodeCatalyst.

## Themen

- [Erweiterungen, Konzepte](#)
- [Erweiterungen installieren und deinstallieren in CodeCatalyst](#)
- [Verwendung von GitHub Repositorys in CodeCatalyst](#)
- [Verwendung von Jira-Issues in CodeCatalyst](#)

## Erweiterungen, Konzepte

Im Folgenden finden Sie einige Konzepte und Begriffe, die Sie kennen sollten, wenn Sie mit Erweiterungen arbeiten CodeCatalyst.

## Erweiterungen

Eine Erweiterung ist ein Add-on, das Sie in Ihrem CodeCatalyst Bereich installieren können, um Ihren Projekten neue Funktionen hinzuzufügen und sie in Dienste außerhalb von zu integrieren CodeCatalyst. Erweiterungen können vom CodeCatalyst Katalog aus durchsucht und installiert werden.

## CodeCatalyst katalog

Der CodeCatalyst Katalog ist eine zentralisierte Liste aller Erweiterungen, die in verfügbar sind CodeCatalyst. Sie können den CodeCatalyst Katalog nach Erweiterungen durchsuchen, die die Erfahrung Ihres Teams in Bereichen CodeCatalyst wie Quellcode, Workflows und mehr verbessern können.

## Erweiterungen installieren und deinstallieren in CodeCatalyst

Eine Erweiterung ist ein Add-on, das Sie in Ihrem CodeCatalyst Bereich installieren können, um Ihren Projekten neue Funktionen hinzuzufügen und sie in Dienste außerhalb von CodeCatalyst zu integrieren. Erweiterungen können vom CodeCatalyst Katalog aus durchsucht und installiert werden.

## Themen

- [Eine Erweiterung installieren](#)
- [Deinstallation einer Erweiterung](#)

## Eine Erweiterung installieren

Sie können Erweiterungen für Ihren CodeCatalyst Bereich installieren, die den Projekten in diesem Bereich Funktionen hinzufügen. Sie können den CodeCatalyst Katalog aufrufen, indem Sie auf das Katalogsymbol

klicken 

### Important

Um eine Erweiterung zu installieren, müssen Sie mit einem Konto angemeldet sein, das in dem Bereich die Rolle eines Space-Administrators innehat.

Um eine Erweiterung aus dem CodeCatalyst Katalog zu installieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Navigiere zum CodeCatalyst Katalog, indem du das Katalogsymbol



im oberen Menü auswählst. Sie können nach Erweiterungen suchen oder Erweiterungen anhand von Kategorien filtern.

4. (Optional) Wählen Sie den Namen der Erweiterung aus, um weitere Informationen zu der Erweiterung zu erhalten, z. B. die Berechtigungen, über die die Erweiterung verfügen wird.
5. Wählen Sie Installieren aus. Überprüfen Sie die für die Erweiterung erforderlichen Berechtigungen. Wenn Sie fortfahren möchten, wählen Sie erneut Installieren aus.

Nach der Installation einer Erweiterung wird die Detailseite für die installierte Erweiterung angezeigt. Weitere Informationen zur Erweiterung finden Sie auf den Tabs. Auf der Detailseite können Sie bei Bedarf auch die weitere Konfiguration der Erweiterung vornehmen.




## Deinstallation einer Erweiterung

Sie können Erweiterungen deinstallieren, die zuvor in Ihrem CodeCatalyst Bereich installiert wurden. Durch die Deinstallation einer Erweiterung werden möglicherweise Ressourcen, die sich auf diese Erweiterung beziehen, aus Ihrem CodeCatalyst Bereich oder Ihren Projekten entfernt.

### Important

Um eine Erweiterung zu deinstallieren, müssen Sie mit einem Konto angemeldet sein, das in dem Bereich die Rolle eines Space-Administrators innehat.

Um eine Erweiterung aus Ihrem CodeCatalyst Space zu deinstallieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Gehen Sie wie folgt vor, um eine Liste der installierten Erweiterungen für Ihren Bereich anzuzeigen:
  - a. Wählen Sie Einstellungen und anschließend Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie bei der Erweiterung, die Sie deinstallieren möchten, die Option Konfigurieren aus.
5. Wählen Sie auf der Detailseite der Erweiterung die Option Deinstallieren aus.
6. Überprüfen Sie die Informationen im Dialogfeld „Erweiterung deinstallieren“. Folgen Sie den Anweisungen und wählen Sie dann Deinstallieren, um die Erweiterung zu deinstallieren.

## Verwendung von GitHub Repositorys in CodeCatalyst

GitHub ist ein cloudbasierter Dienst, der Entwicklern hilft, ihren Code zu speichern und zu verwalten. Mit der GitHub Repository-Erweiterung können Sie verknüpfte GitHub Repositorys in CodeCatalyst Amazon-Projekten verwenden. Sie können GitHub Repositorys auch verknüpfen, wenn Sie ein neues Projekt erstellen. CodeCatalyst Weitere Informationen finden Sie unter [Ein Projekt mit einem verknüpften GitHub Repository erstellen](#).

**Note**

Du kannst leere oder archivierte GitHub Repositories nicht mit CodeCatalyst Projekten verwenden. Die GitHub Repository-Erweiterung ist nicht mit GitHub Enterprise Server kompatibel.

Sobald Sie die GitHub Repository-Erweiterung installiert und konfiguriert haben, können Sie:

- Sehen Sie sich Ihre GitHub Repositories in der Liste der Quell-Repositories an unter CodeCatalyst
- Speichern und verwalten Sie Workflow-Definitionsdateien in Ihren GitHub Repositories.
- Erstellen, lesen, aktualisieren und löschen Sie Dateien, die in verknüpften GitHub Repositories gespeichert sind, aus Entwicklungsumgebungen CodeCatalyst
- Der CodeCatalyst Start-Workflow wird automatisch ausgeführt, wenn Code in Ihre GitHub Repositories übertragen wird
- Verwenden Sie Ihre verknüpften GitHub Repository-Quelldateien in Workflows CodeCatalyst
- Lesen und führen Sie GitHub Aktionen in CodeCatalyst Workflows aus

In den folgenden Themen wird erklärt, wie Sie die GitHub Repository-Erweiterung installieren und konfigurieren und Ihre GitHub Repositories verwenden, sobald sie mit Ihren CodeCatalyst Projekten verknüpft sind.

**Themen**

- [Schnellstart: GitHub Repositories verwenden in CodeCatalyst](#)
- [GitHub Konten verwalten in CodeCatalyst](#)
- [Verwaltung von GitHub Repositorien in CodeCatalyst](#)
- [Verknüpfte GitHub Repositories anzeigen in CodeCatalyst](#)
- [Verwendung von GitHub Repositories in Workflows CodeCatalyst](#)
- [Zugriffsbeschränkungen für IP-Adressen in GitHub Enterprise Cloud verwenden](#)
- [Blockieren von GitHub Pull-Request-Zusammenführungen, wenn Workflows fehlschlagen](#)

## Schnellstart: GitHub Repositorys verwenden in CodeCatalyst

Führen Sie die folgenden Schritte aus, um die GitHub Repository-Erweiterung zu installieren, eine Verbindung zu Ihrem GitHub Konto herzustellen und Ihre GitHub Repositorys mit Ihren vorhandenen CodeCatalyst Projekten zu verknüpfen.

Sie können auch die GitHub Repository-Erweiterung installieren, eine Verbindung zu Ihrem GitHub Konto herstellen und GitHub Repositorys verknüpfen, wenn Sie ein neues Projekt erstellen.

CodeCatalyst Weitere Informationen finden Sie unter [Ein Projekt mit einem verknüpften GitHub Repository erstellen](#).

### Inhalt

- [Schritt 1: Installieren Sie die GitHub Erweiterung aus dem Katalog CodeCatalyst](#)
- [Schritt 2: Connect dein GitHub Konto mit deinem CodeCatalyst Bereich](#)
- [Schritt 3: Verknüpfe dein GitHub Repository mit deinem Projekt CodeCatalyst](#)
- [Nächste Schritte](#)

### Schritt 1: Installieren Sie die GitHub Erweiterung aus dem Katalog CodeCatalyst

Der erste Schritt zur Verwendung von GitHub Repositorys in CodeCatalyst besteht darin, die GitHub Repository-Erweiterung aus dem CodeCatalyst Katalog zu installieren. Um die Erweiterung zu installieren, führen Sie die folgenden Schritte aus und wählen Sie die GitHub Repository-Erweiterung aus.

#### Important

Im Rahmen der Installation und Konfiguration der GitHub Repository-Erweiterung müssen Sie eine Erweiterung in Ihrem GitHub Konto installieren. Dazu müssen Sie sowohl der GitHub Kontoadministrator als auch der CodeCatalyst Space-Administrator sein.

Um eine Erweiterung aus dem CodeCatalyst Katalog zu installieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

3. Navigiere zum CodeCatalyst CodeCatalyst Katalog, indem du das Katalogsymbol



im oberen Menü auswählst. Sie können nach GitHub Repositorys suchen oder Erweiterungen nach Kategorien filtern.

4. (Optional) Wählen Sie den Namen der GitHub Repository-Erweiterung aus, um weitere Informationen über die Erweiterung zu erhalten, z. B. welche Berechtigungen die Erweiterung haben wird.
5. Wählen Sie Installieren aus. Überprüfen Sie die für die Erweiterung erforderlichen Berechtigungen. Wenn Sie fortfahren möchten, wählen Sie erneut Installieren aus.

Nach der Installation der GitHub Repository-Erweiterung werden Sie zur Detailseite der GitHub Repository-Erweiterung weitergeleitet, auf der Sie verbundene GitHub Konten und verknüpfte GitHub Repositorys anzeigen und verwalten können.

## Schritt 2: Connect dein GitHub Konto mit deinem CodeCatalyst Bereich

Nachdem Sie die GitHub Repository-Erweiterung installiert haben, müssen Sie im nächsten Schritt Ihr GitHub Konto mit Ihrem CodeCatalyst Space verbinden.

### Important

Um Ihr GitHub Konto mit Ihrem CodeCatalyst Space zu verbinden, müssen Sie sowohl der GitHub Kontoadministrator als auch der CodeCatalyst Space-Administrator sein.

### Um dein GitHub Konto zu verbinden CodeCatalyst

1. Wählen Sie auf der Registerkarte Verbundene GitHub Konten die Option GitHub Konto Connect aus, für das Sie zur externen Site wechseln möchten GitHub.
2. Melden Sie sich mit Ihren GitHub Anmeldeinformationen bei Ihrem GitHub Konto an und wählen Sie dann das Konto aus, auf dem Sie Amazon installieren möchten CodeCatalyst.

### Tip

Wenn Sie Ihr GitHub Konto zuvor mit einem anderen Bereich verbunden haben, werden Sie nicht aufgefordert, es erneut zu autorisieren. Stattdessen wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, wo Sie die Erweiterung installieren möchten,

wenn Sie Mitglied oder Mitarbeiter in mehr als einem GitHub Konto sind, oder die Konfigurationsseite für die CodeCatalyst Amazon-Anwendung, wenn Sie nur einem GitHub Konto angehören. Konfigurieren Sie die Anwendung für den Repository-Zugriff, den Sie zulassen möchten, und wählen Sie dann Speichern. Wenn die Schaltfläche Speichern in nicht aktiv ist GitHub, nehmen Sie eine Änderung an der Konfiguration vor, und versuchen Sie es dann erneut.

3. Wählen Sie aus, ob Sie den CodeCatalyst Zugriff auf alle aktuellen und future Repositories zulassen möchten, oder wählen Sie die spezifischen GitHub Repositories aus, in denen Sie sie verwenden möchten. CodeCatalyst Die Standardoption ist „Alle GitHub Repositories“ im Konto. GitHub
4. Überprüfen Sie die Berechtigungen, die Sie erhalten haben CodeCatalyst, und wählen Sie dann Installieren.

Nachdem Sie eine Verbindung zu Ihrem GitHub Konto hergestellt haben CodeCatalyst, können Sie das verbundene Konto auf der Detailseite der GitHub Repository-Erweiterung auf der Registerkarte „GitHub Konten“ einsehen.

### Schritt 3: Verknüpfe dein GitHub Repository mit deinem Projekt CodeCatalyst

Der dritte und letzte Schritt bei der Verwendung Ihrer GitHub Repositories CodeCatalyst besteht darin, das Repository mit dem CodeCatalyst Projekt zu verknüpfen, in dem Sie es verwenden möchten.

#### Important

Sie können ein GitHub Repository zwar als Mitwirkender verknüpfen, aber Sie können die Verknüpfung eines GitHub Repositories nur als Space-Administrator oder Projektadministrator aufheben. Weitere Informationen finden Sie unter [Die Verknüpfung eines GitHub Repositories mit einem Projekt aufheben CodeCatalyst](#).

Um ein GitHub Repository auf der Detailseite der GitHub Repository-Erweiterung mit einem CodeCatalyst Projekt zu verknüpfen

1. Wählen Sie auf der Registerkarte Verknüpfte GitHub Repositories die Option Repository verknüpfen GitHub aus.
2. Wählen Sie im GitHub Dropdownmenü für das GitHub Konto das Konto aus, das das Repository enthält, das Sie verknüpfen möchten.

3. Wählen Sie im GitHub Repository-Dropdownmenü das Repository aus, das Sie mit einem CodeCatalyst Projekt verknüpfen möchten.

 Tip

Wenn der Name des Repositorys ausgegraut ist, können Sie dieses Repository nicht verknüpfen, da es bereits mit einem anderen Projekt in dem Bereich verknüpft wurde.

4. Wählen Sie im CodeCatalyst Dropdownmenü für das CodeCatalyst Projekt das Projekt aus, mit dem Sie das GitHub Repository verknüpfen möchten.
5. Wählen Sie Verknüpfen.


Sie können Ihr GitHub Repository auch mit einem Projekt aus den Quell-Repositorys in Code verknüpfen. Weitere Informationen finden Sie unter [Verwaltung von GitHub Repositorien in CodeCatalyst](#).

## Nächste Schritte

Nachdem Sie die GitHub Repository-Erweiterung installiert, Ihr GitHub Konto verbunden und Ihre GitHub Repositorys mit Ihren CodeCatalyst Projekten verknüpft haben, können Sie sie in CodeCatalyst Workflows und Entwicklungsumgebungen verwenden. Weitere Informationen finden Sie unter [Verwendung von GitHub Repositorys in Workflows CodeCatalyst](#) und [Erstellen einer Entwicklungsumgebung](#).

## GitHub Konten verwalten in CodeCatalyst

Um GitHub Repositorys in verwenden zu können CodeCatalyst, musst du dein GitHub Konto mit deinem Bereich verbinden. CodeCatalyst Wenn du die Repositorys eines Accounts nicht mehr verwenden möchtest CodeCatalyst, kannst du die Verbindung zu diesem GitHub Konto trennen. Wenn die Verbindung zu einem Konto getrennt wird, werden durch Ereignisse in den Repositorys des Kontos keine Workflow-Läufe gestartet, und Sie können diese Repositorys nicht mit Entwicklungsumgebungen verwenden. CodeCatalyst

 Important

Um dein GitHub Konto und deinen CodeCatalyst Space zu verbinden oder zu trennen, musst du sowohl der GitHub Kontoadministrator als auch der Space-Administrator sein. CodeCatalyst


## Inhalt

- [Ein GitHub Konto mit einem CodeCatalyst Bereich Connect](#)
- [Während der Projekterstellung ein GitHub Konto mit einem CodeCatalyst Bereich Connect](#)
- [Trennen Sie ein GitHub Konto von einem CodeCatalyst Space](#)

## Ein GitHub Konto mit einem CodeCatalyst Bereich Connect

Führe die folgenden Schritte aus, um ein GitHub Konto mit deinem CodeCatalyst Space zu verbinden.

Um dein GitHub Konto zu verbinden CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Gehen Sie wie folgt vor, um die installierte GitHub Repository-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wählen Sie Einstellungen und anschließend Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in GitHub Repositories die Option Konfigurieren aus.
5. Wählen Sie auf der Registerkarte Verbundene GitHub Konten die Option GitHub Konto Connect aus, für das Sie zur externen Site wechseln möchten GitHub.
6. Melden Sie sich mit Ihren GitHub Anmeldeinformationen bei Ihrem GitHub Konto an und wählen Sie dann das Konto aus, auf dem Sie Amazon installieren möchten CodeCatalyst.

### Tip

Wenn Sie zuvor ein GitHub Konto mit dem Bereich verbunden haben, werden Sie nicht aufgefordert, es erneut zu autorisieren. Stattdessen wird ein Dialogfeld angezeigt, in dem Sie gefragt werden, wo Sie die Erweiterung installieren möchten, wenn Sie Mitglied oder Mitarbeiter in mehr als einem GitHub Bereich sind, oder die Konfigurationsseite für die CodeCatalyst Amazon-Anwendung, wenn Sie nur zu einem GitHub Bereich gehören. Konfigurieren Sie die Anwendung für den Repository-Zugriff, den Sie zulassen

möchten, und wählen Sie dann Speichern. Wenn die Schaltfläche Speichern nicht aktiv ist, nehmen Sie eine Änderung an der Konfiguration vor, und versuchen Sie es dann erneut.

7. Wählen Sie aus, ob Sie den CodeCatalyst Zugriff auf alle aktuellen und future Repositorys zulassen möchten, oder wählen Sie die spezifischen GitHub Repositorys aus, in denen Sie sie verwenden möchten. CodeCatalyst Die Standardoption ist „Alle GitHub Repositorys“ im Bereich. GitHub
8. Überprüfen Sie die Berechtigungen, die Sie erhalten haben CodeCatalyst, und wählen Sie dann Installieren.

Nach der Installation der GitHub Repository-Erweiterung werden Sie zur Detailseite der GitHub Repository-Erweiterung weitergeleitet, auf der Sie verbundene GitHub Konten und verknüpfte GitHub Repositorys anzeigen und verwalten können.

## Während der Projekterstellung ein GitHub Konto mit einem CodeCatalyst Bereich Connect

Du kannst ein GitHub Konto mit deinem CodeCatalyst Bereich verbinden, wenn du ein neues CodeCatalyst Projekt erstellst. Weitere Informationen finden Sie unter [Ein Projekt mit einem verknüpften GitHub Repository erstellen](#).

## Trennen Sie ein GitHub Konto von einem CodeCatalyst Space


Gehen Sie wie folgt vor, um ein GitHub Konto von Ihrem CodeCatalyst Space zu trennen. Sobald die Verbindung zu einem Konto getrennt wurde, werden durch Ereignisse in den Repositorys des Kontos keine Workflow-Läufe gestartet, und Sie können diese Repositorys nicht mit CodeCatalyst Entwicklungsumgebungen verwenden.

### Note

Um die Verbindung zu einem GitHub Konto zu trennen, müssen Sie zunächst die Verknüpfung aller verknüpften GitHub Repositorys mit diesem Konto aufheben. Weitere Informationen finden Sie unter [Die Verknüpfung eines GitHub Repositorys mit einem Projekt aufheben CodeCatalyst](#).



## Um die Verbindung zu einem Konto zu trennen GitHub

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Gehen Sie wie folgt vor, um die installierte GitHub Repository-Erweiterung für Ihren CodeCatalyst Bereich anzuzeigen:
  - a. Wählen Sie Einstellungen und anschließend Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in GitHub Repositories die Option Konfigurieren aus.
5. Wählen Sie auf der Registerkarte GitHub Konten das GitHub Konto aus, dessen Verbindung Sie trennen möchten.
6. Wählen Sie GitHub Konto trennen.
7. Prüfen Sie im Dialogfeld „Verbindung trennen“, wie sich die Trennung des Kontos auswirkt.
8. Geben Sie Disconnect in das Texteingabefeld ein und wählen Sie dann Disconnect aus.

## Verwaltung von GitHub Repositorien in CodeCatalyst

Um GitHub Repositories in verwenden zu können CodeCatalyst, müssen Sie die Repositories zunächst mit Ihrem Projekt verknüpfen. CodeCatalyst Bevor Sie ein GitHub Repository verknüpfen können, müssen Sie das GitHub Konto, zu dem das Repository gehört, mit Ihrem CodeCatalyst Space verbinden. Weitere Informationen finden Sie unter [Ein GitHub Konto mit einem CodeCatalyst Bereich Connect](#).

Wenn Sie das GitHub Repository in nicht mehr verwenden möchten CodeCatalyst, können Sie die Verknüpfung mit einem CodeCatalyst Projekt aufheben. Wenn die Verknüpfung eines Repositories aufgehoben wird, werden durch Ereignisse in diesem Repository keine Workflow-Ausführungen gestartet, und Sie können dieses Repository nicht mit CodeCatalyst Entwicklungsumgebungen verwenden.

### Inhalt

- [Verknüpfen Sie ein GitHub Repository mit einem Projekt CodeCatalyst](#)

- [Verknüpfen Sie während der Projekterstellung ein GitHub Repository mit einem CodeCatalyst Projekt](#)
- [Die Verknüpfung eines GitHub Repositorys mit einem Projekt aufheben CodeCatalyst](#)


## Verknüpfen Sie ein GitHub Repository mit einem Projekt CodeCatalyst

Sie können verknüpfte GitHub Repositories in Workflows verwenden, bei denen Ereignisse im verknüpften GitHub Repository Workflows starten, die je nach Workflow-Konfiguration Code erstellen, testen oder bereitstellen können. Workflow-Konfigurationsdateien für Workflows, die verknüpfte GitHub Repositories verwenden, werden im GitHub verknüpften Repository gespeichert. Verknüpfte GitHub Repositories können auch mit Entwicklungsumgebungen verwendet werden, um Dateien im verknüpften GitHub Repository zu erstellen, zu aktualisieren und zu löschen. Sie können ein GitHub Repository entweder über die Detailseite der Repository-Erweiterung oder über die GitHub Quell-Repository-Ansicht unter Code im CodeCatalyst Projekt selbst mit einem Projekt verknüpfen.

### Note

Ein GitHub Repository kann nur mit einem CodeCatalyst Projekt in einem Space verknüpft werden. Sie können kein Repository verknüpfen, das denselben Namen wie ein anderes Repository im Projekt hat.

Um ein GitHub Repository von der Detailseite der GitHub Repository-Erweiterung aus mit einem CodeCatalyst Projekt zu verknüpfen


1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Gehen Sie wie folgt vor, um die installierte GitHub Repository-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wählen Sie Einstellungen und anschließend Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in GitHub Repositories die Option Konfigurieren aus.

5. Wählen Sie auf der Registerkarte Verknüpfte GitHub Repositories die Option Repository verknüpfen GitHub aus.
6. Wählen Sie aus der GitHub Konto-Dropdown-Liste das GitHub Konto aus, das das Repository enthält, das Sie verknüpfen möchten.
7. Wählen Sie aus der GitHub Repository-Dropdownliste das Repository aus, das Sie mit einem CodeCatalyst Projekt verknüpfen möchten.

 Tip

Wenn der Name des Repositories ausgegraut ist, können Sie dieses Repository nicht verknüpfen, da es bereits mit einem anderen Projekt in dem Bereich verknüpft wurde.

8. (Optional) Wenn Sie in der Liste der Repositories kein GitHub Repository sehen, wurde es möglicherweise nicht für den Repository-Zugriff in der CodeCatalyst Amazon-Anwendung in GitHub konfiguriert. Sie können im verbundenen Konto konfigurieren, CodeCatalyst in welchen GitHub Repositories verwendet werden können.
  - a. Navigieren Sie zu Ihrem [GitHub](#)Konto, wählen Sie Einstellungen und dann Anwendungen.
  - b. Wählen Sie auf der Registerkarte Installierte GitHub Apps die Option Konfigurieren für die CodeCatalyst Amazon-Anwendung aus.
  - c. Gehen Sie wie folgt vor, um den Zugriff auf die GitHub Repositories zu konfigurieren, die Sie verknüpfen möchten: CodeCatalyst
    - Um Zugriff auf alle aktuellen und future Repositories zu gewähren, wählen Sie Alle Repositories.
    - Um Zugriff auf bestimmte Repositories zu gewähren, wählen Sie Nur ausgewählte Repositories aus, wählen Sie das Drop-down-Menü Repositories auswählen und wählen Sie dann ein Repository aus, zu dem Sie Links zulassen möchten. CodeCatalyst

 Note

- Du kannst leere oder archivierte GitHub Repositories nicht mit Projekten verwenden. CodeCatalyst
- Sie können ein GitHub Repository, das denselben Namen wie ein Repository in einem CodeCatalyst Projekt hat, nicht verknüpfen.

- Die GitHub Repository-Erweiterung ist nicht mit GitHub Enterprise Server-Repositories kompatibel.

9. Wählen Sie im Dropdownmenü für das CodeCatalyst Projekt das CodeCatalyst Projekt aus, mit dem Sie das GitHub Repository verknüpfen möchten.
10. Wählen Sie Verknüpfen.

Um ein GitHub Repository von der Seite mit den Quell-Repositories in einem CodeCatalyst Projekt aus einem Projekt zu verknüpfen


1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem CodeCatalyst Projekt.
3. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.
4. Wählen Sie Repository hinzufügen und anschließend Repository verknüpfen aus.
5. Wählen Sie im Dropdownmenü Repository-Anbieter die Option GitHub.
6. Wählen Sie im GitHub Dropdownmenü für das GitHub Konto das Konto aus, das das Repository enthält, das Sie verknüpfen möchten.
7. Wählen Sie im GitHub Repository-Dropdownmenü das GitHub Repository aus, das Sie mit Ihrem CodeCatalyst Projekt verknüpfen möchten.

 Tip

Wenn der Name des Repositories ausgegraut ist, können Sie dieses Repository nicht verknüpfen, da es bereits mit einem anderen Projekt in Amazon CodeCatalyst verknüpft wurde.

8. (Optional) Wenn Sie in der GitHub Repository-Dropdownliste kein GitHub Repository sehen, wurde es möglicherweise nicht für den Repository-Zugriff in der Space-Anwendung in konfiguriert. GitHub Sie können im verbundenen Konto konfigurieren, CodeCatalyst in welchen GitHub Repositories verwendet werden können.
  - a. Navigieren Sie zu Ihrem [GitHub](#)Konto, wählen Sie Einstellungen und dann Anwendungen.
  - b. Wählen Sie auf der Registerkarte Installierte GitHub Apps die Option Konfigurieren für die CodeCatalyst Amazon-Anwendung aus.

- c. Gehen Sie wie folgt vor, um den Zugriff auf die GitHub Repositories zu konfigurieren, die Sie verknüpfen möchten: CodeCatalyst
  - Um Zugriff auf alle aktuellen und future Repositories zu gewähren, wählen Sie Alle Repositories.
  - Um Zugriff auf bestimmte Repositories zu gewähren, wählen Sie Nur ausgewählte Repositories aus, wählen Sie das Drop-down-Menü Repositories auswählen und wählen Sie dann ein Repository aus, zu dem Sie Links zulassen möchten. CodeCatalyst

 Note

- Du kannst leere oder archivierte GitHub Repositories nicht mit Projekten verwenden. CodeCatalyst
- Sie können ein GitHub Repository, das denselben Namen wie ein Repository in einem CodeCatalyst Projekt hat, nicht verknüpfen.
- Die GitHub Repository-Erweiterung ist nicht mit GitHub Enterprise Server-Repositories kompatibel.

## 9. Wählen Sie Verknüpfen.

### Verknüpfen Sie während der Projekterstellung ein GitHub Repository mit einem CodeCatalyst Projekt

Sie können ein GitHub Repository mit einem neuen CodeCatalyst Projekt verknüpfen, wenn Sie das neue CodeCatalyst Projekt erstellen. Weitere Informationen finden Sie unter [Ein Projekt mit einem verknüpften GitHub Repository erstellen](#).


### Die Verknüpfung eines GitHub Repositories mit einem Projekt aufheben CodeCatalyst

Durch das Aufheben der Verknüpfung mit einem Repository wird das GitHub Repository nicht gelöscht und es werden keine Änderungen daran vorgenommen. Es werden keine Workflow-Konfigurationsdateien gelöscht, die in dem verknüpften Repository gespeichert sind. Sobald Sie jedoch die Verknüpfung mit einem GitHub Repository aufheben, werden durch Ereignisse in diesem Repository keine Workflow-Ausführungen mehr gestartet, und Sie können das Repository nicht mit Entwicklungsumgebungen verwenden.

**⚠ Important**

Um die Verknüpfung zwischen Ihrem GitHub Repository und Ihrem CodeCatalyst Projekt aufzuheben, müssen Sie der Space-Administrator oder der Projektadministrator sein.

Um die Verknüpfung eines Repositorys aufzuheben GitHub

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Gehen Sie wie folgt vor, um die installierte GitHub Repository-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wählen Sie Einstellungen und anschließend Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in GitHub Repositorys die Option Konfigurieren aus.
5. Wählen Sie auf der Registerkarte Verknüpfte GitHub Repositorys das GitHub Repository aus, dessen Verknüpfung Sie aufheben möchten.
6. Wählen Sie Repository-Verknüpfung aufheben GitHub .
7. Prüfen Sie im Dialogfeld „Verknüpfung aufheben“, welche Auswirkungen das Aufheben der Verknüpfung mit dem Repository hat.
8. Geben Sie unlink in das Texteingabefeld ein und wählen Sie Unlink aus.

## Verknüpfte GitHub Repositorys anzeigen in CodeCatalyst

Sie können die verknüpften GitHub Repositorys in der Liste der Quell-Repositorys für Ihr Projekt oder auf der Detailseite der GitHub Repository-Erweiterung einsehen. Wenn Sie sie aus der Liste der Repositorys auswählen, werden sie nicht geöffnet. CodeCatalyst Stattdessen werden sie geöffnet GitHub, wo Sie den Code im verknüpften Repository ansehen und bearbeiten können.

Um verknüpfte GitHub Repositorys anzuzeigen in CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

2. Navigieren Sie zu Ihrem CodeCatalyst Projekt.
3. Wählen Sie im Navigationsbereich Code und dann Quell-Repositories aus.

Um verknüpfte GitHub Repositories von Ihnen aus anzuzeigen: die Seite mit den Erweiterungsdetails

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem CodeCatalyst Bereich und wählen Sie dann den Tab Installierte Erweiterungen aus.
3. Wählen Sie unter GitHub Repositories die Option Konfigurieren aus.
4. Wählen Sie den Tab Verknüpfte GitHub Repositories, um alle GitHub Repositories anzuzeigen, die mit CodeCatalyst Projekten in Ihrem Bereich verbunden sind. CodeCatalyst

Die GitHub Repositories, die mit deinem Projekt verknüpft sind, werden in der Liste angezeigt. Wählen Sie das GitHub Repository aus, in GitHub dem Sie Dateien anzeigen und bearbeiten möchten.

#### Note

Wenn ein Workflow ein GitHub Repository in einer Quellaktion verwendet, werden Änderungen, die Sie am Workflow-YAML im Visual Editor oder im YAML-Editor vornehmen, CodeCatalyst automatisch festgeschrieben und in das GitHub Repository übertragen.

## Verwendung von GitHub Repositories in Workflows CodeCatalyst

Sie können ein verknüpftes GitHub Repository als Quelle für einen Workflow verwenden, bei dem Änderungen an einem bestimmten Zweig in einem verknüpften GitHub Repository automatisch eine Workflow-Ausführung starten.

Ein Workflow ist ein automatisiertes Verfahren, das beschreibt, wie Sie Ihren Code als Teil eines CI/CD-Systems (Continuous Integration and Continuous Delivery) erstellen, testen und bereitstellen. Ein Workflow definiert eine Reihe von Schritten oder Aktionen, die während einer Workflow-Ausführung ausgeführt werden sollen. Ein Workflow definiert auch die Ereignisse oder Auslöser, die den Start des Workflows auslösen. Um einen Workflow einzurichten, erstellen Sie mit dem [visuellen Editor oder dem YAML-Editor](#) der CodeCatalyst Konsole eine Workflow-Definitionsdatei.

**Tip**

Um einen kurzen Überblick darüber zu erhalten, wie Sie Workflows in einem Projekt verwenden könnten, [erstellen Sie ein Projekt mit einem Blueprint](#). Jeder Blueprint stellt einen funktionierenden Workflow bereit, den Sie überprüfen, ausführen und mit dem Sie experimentieren können.

Wenn Sie einen Workflow für die Verwendung eines verknüpften GitHub Repositorys konfigurieren, wird die Workflow-Konfigurationsdatei in diesem GitHub Repository gespeichert. Die Workflow-Konfiguration ist eine YAML-Datei, die den Workflow-Namen, die Trigger, Ressourcen, Artefakte und Aktionen definiert. Weitere Informationen zur Workflow-Konfigurationsdatei finden Sie unter [Referenz zur Workflow-Definition](#).

Die Workflow-Konfigurationsdatei muss sich im `./codecatalyst/workflows/` Verzeichnis Ihres GitHub Repositorys befinden.

Sie können den Workflow-Editor verwenden, um Workflows zu erstellen und zu konfigurieren. Weitere Informationen finden Sie unter [Erste Schritte mit Workflows in CodeCatalyst](#) und [Mit Quellen arbeiten](#).

## Startet automatisch eine Workflow-Ausführung nach GitHub Repository-Ereignissen

Sie können einen CodeCatalyst Workflow so konfigurieren, dass er automatisch eine Ausführung startet, wenn Code in den angegebenen Zweig Ihres GitHub Repositorys übertragen wird. Um eine Workflow-Ausführung automatisch zu starten, fügen Sie dem `Triggers` Abschnitt der Workflow-Konfigurationsdatei einen Trigger hinzu.

Beispiel: Ein einfacher Code-Push-Trigger

Das folgende Beispiel zeigt einen Trigger, der eine Workflow-Ausführung startet, wenn Code in einen beliebigen Branch in Ihrem Quell-Repository übertragen wird.

```
Triggers:  
- Type: PUSH
```

Beispiel: Ein einfacher Pull-Request-Trigger

Das folgende Beispiel zeigt einen Trigger, der einen Workflow-Lauf startet, wenn ein Pull-Request für einen Branch in Ihrem Quell-Repository erstellt wird.



**Triggers:**

- Type: PULLREQUEST

**Events:**

- OPEN

Weitere Informationen finden Sie unter [Arbeiten mit Auslösern](#).

## Zugriffsbeschränkungen für IP-Adressen in GitHub Enterprise Cloud verwenden

Die CodeCatalyst GitHub Amazon-Repository-Erweiterung ist mit den [IP-Zugriffsbeschränkungen GitHub für Enterprise Cloud](#) kompatibel. Wenn Sie eine GitHub Enterprise Cloud-Organisation so konfigurieren, dass der Zugriff auf bestimmte IP-Adressen beschränkt wird, können Sie [GitHub Apps auch zur Konfiguration der Zulassungsliste aktivieren](#), sodass IP-Adressen automatisch CodeCatalyst registriert werden können. Alternativ können Sie [die CodeCatalyst IP-Adressen manuell hinzufügen](#). Weitere Informationen finden Sie unter [IP-Adressen, die von der GitHub Repository-Erweiterung verwendet werden](#).

Wenn die CodeCatalyst IP-Adressen nicht in der Zulassungsliste einer GitHub Organisation stehen, kann die CodeCatalyst GitHub Amazon-App nicht auf Ihre GitHub Repositories zugreifen.

### IP-Adressen, die von der GitHub Repository-Erweiterung verwendet werden

Die folgenden IP-Adressen werden von der GitHub Repository-Erweiterung für den Zugriff auf die Ressourcen Ihrer GitHub Organisation verwendet:

**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

## Blockieren von GitHub Pull-Request-Zusammenführungen, wenn Workflows fehlschlagen

Nachdem du ein GitHub Repository mit verknüpft hast CodeCatalyst, kannst du CodeCatalyst Workflows für Pull Requests hinzufügen. Eine oder mehrere Workflow-Ausführungen können für einen bestimmten Commit erfolgen, und der Ausführungsstatus jedes Workflows in CodeCatalyst wird auch als Teil des Commit-Status in wiedergegeben GitHub. Wenn ein neuer Commit übertragen wird, werden neue [Workflow-Ausführungsstatus](#) GitHub für diesen neuen Commit angezeigt. Wenn Sie einen Workflow für einen Commit erneut ausführen, überschreibt der neue Workflow-Ausführungsstatus den vorherigen Status für diesen Commit und diesen Workflow.

Sie können Branch-Schutzregeln so einrichten GitHub , dass die Zusammenführung von Pull-Requests blockiert wird, wenn der letzte Commit den Status Workflow-Ausführung fehlgeschlagen ist. Bei Branch-Schutzregeln wirkt sich der Status des letzten Commits auf die Fähigkeit aus, eine Pull-Anfrage zusammenzuführen GitHub. Weitere Informationen finden Sie GitHub in den Dokumentationen [Über Statuschecks](#) und [Über geschützte Branches](#). Weitere Informationen zu Workflows finden Sie unter [Mit Läufen arbeiten](#) und [Arbeiten mit Auslösern](#).

## Verwendung von Jira-Issues in CodeCatalyst

Jira ist eine Softwareanwendung, die agilen Entwicklungsteams dabei unterstützt, Arbeit zu planen, zuzuweisen, nachzuverfolgen, zu melden und zu verwalten. Mit der Jira Software-Erweiterung können Sie Jira-Projekte in CodeCatalyst Amazon-Projekten verwenden.

### Note

CodeCatalyst ist nur mit Jira Software Cloud kompatibel.

Sobald Sie die Jira Software-Erweiterung für ein CodeCatalyst Amazon-Projekt installiert und konfiguriert haben, können Sie:

1. Greifen Sie auf Jira-Projekte zu, CodeCatalyst indem Sie sie mit Projekten verknüpfen CodeCatalyst
2. Aktualisiere Jira-Probleme mit CodeCatalyst Pull-Requests
3. Sieh dir den Status und die Workflow-Ausführungen verknüpfter CodeCatalyst Pull-Requests in Jira-Issues an

In den folgenden Themen wird erklärt, wie Sie die Jira Software-Erweiterung für die Verwendung von Jira installieren und konfigurieren. CodeCatalyst

## Themen

- [Schnellstart: Jira-Probleme verwenden in CodeCatalyst](#)
- [Verwaltung von Jira-Seiten in CodeCatalyst](#)
- [Verwaltung von Jira-Projekten in CodeCatalyst](#)
- [Ein Jira-Issue mit einer CodeCatalyst Pull-Anfrage verknüpfen](#)
- [CodeCatalyst Ereignisse in Jira anzeigen](#)
- [Jira-Probleme suchen in CodeCatalyst](#)

## Schnellstart: Jira-Probleme verwenden in CodeCatalyst

Führen Sie die folgenden Schritte aus, um die Jira Software-Erweiterung zu installieren, Ihre Jira-Sites mit Ihren CodeCatalyst Spaces zu verbinden und Ihre Jira-Projekte mit Ihren Projekten zu verknüpfen. CodeCatalyst

### Inhalt

- [Schritt 1: Installieren Sie die Jira Software-Erweiterung aus dem Katalog CodeCatalyst](#)
- [Schritt 2: Connect deine Jira-Seiten mit deinem Space CodeCatalyst](#)
- [Schritt 3: Verknüpfen Sie Ihr Jira-Projekt mit Ihrem Projekt CodeCatalyst](#)
- [Schritt 4: Deine Jira-Issues mit einer Pull-Anfrage verknüpfen CodeCatalyst](#)
- [Nächste Schritte](#)

## Schritt 1: Installieren Sie die Jira Software-Erweiterung aus dem Katalog CodeCatalyst


Der erste Schritt zur Verwaltung von Jira in CodeCatalyst ist die Installation der Jira Software-Erweiterung aus dem Katalog. CodeCatalyst Um die Erweiterung zu installieren, führen Sie die folgenden Schritte aus und wählen Sie die Jira Software-Erweiterung aus.

### Important

Im Rahmen der Installation und Konfiguration der Jira Software-Erweiterung benötigst du ein Atlassian-Konto und eine bestehende Jira-Site. Ein Jira-Projekt kann jedoch zu einem

späteren Zeitpunkt erstellt werden. Du musst der Administrator der Jira-Site und der Space-Administrator sein. CodeCatalyst

Um eine Erweiterung aus dem CodeCatalyst Katalog zu installieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Navigiere zum CodeCatalyst CodeCatalyst Katalog, indem du das Katalogsymbol  im oberen Menü auswählst. Du kannst nach Jira Software suchen oder Erweiterungen nach Kategorien filtern.
4. (Optional) Wähle den Namen der Jira Software-Erweiterung aus, um weitere Informationen zur Erweiterung zu erhalten, z. B. welche Berechtigungen die Erweiterung haben wird.
5. Wählen Sie Installieren aus. Überprüfe die für die Erweiterung erforderlichen Berechtigungen. Wenn du fortfahren möchtest, wähle erneut Installieren.

Nach der Installation der Jira Software-Erweiterung werden Sie zur Detailseite der Jira Software-Erweiterung weitergeleitet, auf der Sie verbundene Jira-Sites und verknüpfte Jira-Projekte anzeigen und verwalten können.

## Schritt 2: Connect deine Jira-Seiten mit deinem Space CodeCatalyst

Nachdem Sie die Jira Software-Erweiterung installiert haben, besteht der nächste Schritt darin, Ihre Jira-Site mit Ihrem Space zu verbinden. CodeCatalyst


### Important

Um Ihre Jira-Site mit Ihrem CodeCatalyst Space zu verbinden, müssen Sie sowohl der Jira-Site-Administrator als auch der Space-Administrator sein. CodeCatalyst

Um deine Jira-Site zu verbinden CodeCatalyst

1. Wähle im Tab Verbundene Jira-Sites die Option Jira-Site Connect aus, um zur externen Site für Atlassian Marketplace zu wechseln.

2. Wähle Jetzt herunterladen, um mit der Installation CodeCatalyst auf deiner Jira-Site zu beginnen.
3. Führen Sie je nach Ihrer Rolle einen der folgenden Schritte aus:
  1. Wenn Sie ein Jira-Site-Administrator sind, wählen Sie im Dropdownmenü der Site die Jira-Site aus, auf der die CodeCatalyst Anwendung installiert werden soll, und wählen Sie dann App installieren aus.

 Note


Wenn Sie eine Jira-Site haben, wird dieser Schritt nicht angezeigt und Sie werden automatisch zum nächsten Schritt weitergeleitet.

2. a. Wenn Sie kein Jira-Administrator sind, wählen Sie im Dropdownmenü der Site die Jira-Site aus, auf der die CodeCatalyst Anwendung installiert werden soll, und wählen Sie dann App anfordern aus. Weitere Informationen zur Installation von Jira-Apps findest du unter [Wer kann Apps installieren?](#) .
  - b. Geben Sie den Grund für die Installation CodeCatalyst in das Textfeld ein oder behalten Sie den Standardtext bei und wählen Sie dann Anfrage abschicken.
4. Überprüfen Sie die Aktionen, die CodeCatalyst bis zur Installation der Anwendung ausgeführt wurden, und wählen Sie dann Jetzt herunterladen.
5. Nachdem die Anwendung installiert ist, wählen Sie Zurück zu, CodeCatalyst um zu zurückzukehren CodeCatalyst.

Nachdem Sie Ihre Jira-Site mit verbunden haben CodeCatalyst, können Sie die verbundene Site auf der Detailseite der Jira Software-Erweiterung im Tab Verbundene Jira-Sites einsehen.

### Schritt 3: Verknüpfen Sie Ihr Jira-Projekt mit Ihrem Projekt CodeCatalyst

Der dritte Schritt zur Verwaltung Ihres Jira-Projekts CodeCatalyst besteht darin, das Jira-Projekt mit dem Projekt zu verknüpfen, in dem Sie es verwenden möchten. CodeCatalyst

 Note

Ein CodeCatalyst Projekt kann nur mit einem Jira-Projekt verknüpft werden. Ein Jira-Projekt kann mit mehreren CodeCatalyst Projekten verknüpft werden.

**⚠ Important**

Um Ihr Jira-Projekt mit Ihrem CodeCatalyst Projekt zu verknüpfen, müssen Sie der CodeCatalyst Space-Administrator oder CodeCatalyst Projektadministrator sein.

Um ein Jira-Projekt auf der Detailseite der Jira Software-Erweiterung mit einem CodeCatalyst Projekt zu verknüpfen

1. Wählen Sie auf der Registerkarte Verknüpfte Jira-Projekte die Option Jira-Projekt verknüpfen aus.
2. Wählen Sie im Dropdownmenü Jira-Site die Jira-Site aus, die das Projekt enthält, das Sie verknüpfen möchten.
3. Wählen Sie im Dropdownmenü für das Jira-Projekt das Projekt aus, das Sie mit einem Projekt verknüpfen möchten. CodeCatalyst
4. Wählen Sie im CodeCatalyst Drop-down-Menü für das CodeCatalyst Projekt das Projekt aus, das Sie mit einem Jira-Projekt verknüpfen möchten.
5. Wählen Sie Verknüpfen.

Sobald ein Jira-Projekt mit einem CodeCatalyst Projekt verknüpft ist, ist der Zugriff auf CodeCatalyst Issues vollständig deaktiviert und Issues im CodeCatalyst Navigationsbereich wird durch ein Jira-Issues-Element ersetzt, das auf das Jira-Projekt verweist.

#### Schritt 4: Deine Jira-Issues mit einer Pull-Anfrage verknüpfen CodeCatalyst

Nachdem du ein Jira-Projekt mit einem Projekt verknüpft hast, kannst du CodeCatalyst Issues verknüpfen, indem du einen Pull Request erstellst und sie als Eigenschaft des Pull-Requests anzeigen lässt. CodeCatalyst

**ℹ Note**

Du kannst keinen Pull Request ohne ein Quell-Repository mit zwei Branches in deinem CodeCatalyst Projekt erstellen. Weitere Informationen zu Pull Requests findest du unter [Arbeiten mit Pull Requests in CodeCatalyst](#).

## Um Jira-Issues mit einem CodeCatalyst Pull Request zu verknüpfen

1. Wählen Sie im Navigationsbereich Code und dann Pull Requests aus.
2. Wählen Sie Pull-Request erstellen, um die Pull-Request-Details einzugeben.
3. Wählen Sie im Drop-down-Menü Quell-Repository das Quell-Repository aus, mit dem Sie einen Pull-Request verknüpfen möchten.
4. Wählen Sie im Drop-down-Menü Quell-Branch den Branch aus, der die Änderungen enthält, die Sie überprüfen möchten.
5. Wählen Sie im Dropdownmenü Zielzweig den Zweig aus, in dem Sie die überprüften Änderungen zusammenführen möchten.
6. Geben Sie im Texteingabefeld Titel der Pull-Anfrage den Titel Ihrer Pull-Requests ein.
7. Wähle „Probleme verknüpfen“ für die Jira-Probleme — optionales Feld aus, wähle das Drop-down-Menü und suche nach den Jira-Vorgängen, die du aus dem verknüpften Jira-Projekt hinzufügen möchtest.
8. Wählen Sie die Jira-Issues aus, die Sie der Pull-Anfrage hinzufügen möchten.
9. Wähle Erstellen, um den Pull-Request zu erstellen.

Der zusammenfassende Status des Pull-Requests und der Status der zugehörigen CodeCatalyst Workflow-Ereignisse spiegeln sich in Ihrem Jira-Issue wider.

## Nächste Schritte

Nachdem Sie die Jira Software-Erweiterung installiert, Ihre Jira-Site verbunden, Ihre Jira-Projekte mit Ihrem CodeCatalyst Projekt verknüpft und eine Pull-Anfrage verknüpft haben, CodeCatalyst werden Updates von in Ihrem Jira-Projekt wiedergegeben. Weitere Informationen zum Anzeigen von CodeCatalyst Ereignissen in Jira finden Sie unter. [CodeCatalyst Ereignisse in Jira anzeigen](#)

## Verwaltung von Jira-Seiten in CodeCatalyst

Um Jira-Projekte in verwalten zu können CodeCatalyst, müssen Sie Ihre Jira-Sites mit Ihrem Bereich verbinden. CodeCatalyst Wenn Sie die Projekte einer Jira-Site nicht mehr verwenden möchten CodeCatalyst, können Sie die Verbindung zu dieser Jira-Site trennen. Wenn die Verbindung zu einer Jira-Site unterbrochen wird, sind Jira-Issues aus den Projekten der Site nicht in den CodeCatalyst Projekten verfügbar und CodeCatalyst Issues wird wieder als Problemquelle verwendet.

**⚠ Important**

Um deine Jira-Site und deinen CodeCatalyst Space zu verbinden oder zu trennen, musst du sowohl der Jira-Site-Administrator als auch der Space-Administrator sein. CodeCatalyst


**Inhalt**

- [Eine Jira-Site mit einem Space verbinden CodeCatalyst](#)
- [Verbindung zwischen einer Jira-Site und einem Bereich trennen CodeCatalyst](#)

**Eine Jira-Site mit einem Space verbinden CodeCatalyst**

Führen Sie die folgenden Schritte aus, um eine Jira-Site mit Ihrem CodeCatalyst Space zu verbinden.

Um deine Jira-Site zu verbinden CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Führen Sie einen der folgenden Schritte aus, um die installierte Jira Software-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wähle Einstellungen und dann Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in Jira Software die Option Konfigurieren aus.
5. Wähle im Tab Verbundene Jira-Sites die Option Jira-Site Connect aus, um zur externen Site für Atlassian Marketplace zu wechseln.
6. Wähle Jetzt herunterladen, um mit der Installation CodeCatalyst auf deiner Jira-Site zu beginnen.


**📘 Note**

Wenn Sie zuvor CodeCatalyst auf Ihrer Jira-Site installiert haben, werden Sie benachrichtigt. Wählen Sie Erste Schritte, um zum letzten Schritt zu gelangen.

7. Führen Sie je nach Ihrer Rolle einen der folgenden Schritte aus:



1. Wenn Sie ein Jira-Site-Administrator sind, wählen Sie im Dropdownmenü der Site die Jira-Site aus, auf der die CodeCatalyst Anwendung installiert werden soll, und wählen Sie dann App installieren aus.

 Note


Wenn Sie eine Jira-Site haben, wird dieser Schritt nicht angezeigt und Sie werden automatisch zum nächsten Schritt weitergeleitet.

2. a. Wenn Sie kein Jira-Administrator sind, wählen Sie im Dropdownmenü der Site die Jira-Site aus, auf der die CodeCatalyst Anwendung installiert werden soll, und wählen Sie dann App anfordern aus. Weitere Informationen zur Installation von Jira-Apps findest du unter [Wer kann Apps installieren?](#) .
  - b. Geben Sie den Grund für die Installation CodeCatalyst in das Textfeld ein, oder behalten Sie den Standardtext bei, und wählen Sie dann Anfrage abschicken.
8. Überprüfen Sie die Aktionen, die CodeCatalyst bis zur Installation der Anwendung ausgeführt wurden, und wählen Sie dann Jetzt herunterladen.
9. Nachdem die Anwendung installiert ist, wählen Sie Zurück zu, CodeCatalyst um zu zurückzukehren CodeCatalyst.

Nachdem Sie Ihre Jira-Site mit verbunden haben CodeCatalyst, können Sie die verbundene Site auf der Detailseite der Jira Software-Erweiterung im Tab Verbundene Jira-Sites einsehen.

## Verbindung zwischen einer Jira-Site und einem Bereich trennen CodeCatalyst


Gehen Sie wie folgt vor, um eine Jira-Site von Ihrem Space zu trennen. CodeCatalyst

 Note

Um die Verbindung zu einer Jira-Site zu trennen, müssen Sie zunächst die Verknüpfung aller verknüpften Jira-Projekte mit diesem Konto aufheben. Weitere Informationen finden Sie unter [Die Verknüpfung eines Jira-Projekts mit einem Projekt aufheben CodeCatalyst](#).

Um die Verbindung zu einer Jira-Site zu trennen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.

2. Navigiere zu deinem CodeCatalyst Bereich.
3. Führen Sie einen der folgenden Schritte aus, um die installierte Jira Software-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wähle Einstellungen und dann Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in Jira Software die Option Konfigurieren aus.
5. Wählen Sie auf der Registerkarte Verbundene Jira-Sites die Jira-Site aus, deren Verbindung Sie trennen möchten.
6. Wählen Sie Jira-Site trennen.
7. Prüfen Sie im Dialogfeld „Verbindung trennen“, wie sich das Trennen der Verbindung zur Site auswirkt.
8. Geben Sie Disconnect in das Texteingabefeld ein und wählen Sie dann Disconnect aus.

## Verwaltung von Jira-Projekten in CodeCatalyst

Um ein Jira-Projekt in zu verwalten CodeCatalyst, müssen Sie das Jira-Projekt zunächst mit Ihrem Projekt verknüpfen. CodeCatalyst Anschließend können Sie Jira-Issues zu einer CodeCatalyst Pull-Anfrage in einem Projekt hinzufügen und verwenden. CodeCatalyst Bevor Sie ein Jira-Projekt verknüpfen können, müssen Sie die Jira-Site, zu der das Projekt gehört, mit Ihrem Bereich verbinden. CodeCatalyst Weitere Informationen finden Sie unter [Eine Jira-Site mit einem Space verbinden CodeCatalyst](#).

Wenn Sie ein Jira-Projekt in nicht mehr verwenden möchten CodeCatalyst, können Sie die Verknüpfung mit Ihrem Projekt aufheben. CodeCatalyst Wenn die Verknüpfung eines Jira-Projekts aufgehoben wird, sind Jira-Issues im CodeCatalyst Projekt nicht mehr verfügbar und CodeCatalyst Issues sind wieder der Problemgeber.

### Important

Um Ihr Jira-Projekt mit Ihrem Projekt zu verknüpfen, müssen Sie der CodeCatalyst Space-Administrator oder CodeCatalyst Projektadministrator sein. CodeCatalyst

## Inhalt

- [Ein Jira-Projekt mit einem Projekt verknüpfen CodeCatalyst](#)
- [Die Verknüpfung eines Jira-Projekts mit einem Projekt aufheben CodeCatalyst](#)


## Ein Jira-Projekt mit einem Projekt verknüpfen CodeCatalyst

Du kannst verknüpfte Jira-Projekte verwenden, um Issues zu verwalten und CodeCatalyst Pull Requests mit einem Jira-Issue zu verknüpfen. Der zusammenfassende Status einer Pull-Anfrage und der Status der zugehörigen CodeCatalyst Workflow-Ereignisse werden in Ihrem Jira-Issue wiedergegeben.

### Note

Ein CodeCatalyst Projekt kann nur mit einem Jira-Projekt verknüpft werden. Ein Jira-Projekt kann mit mehreren CodeCatalyst Projekten verknüpft werden.

Um ein Jira-Projekt auf der Detailseite der Jira Software-Erweiterung mit einem CodeCatalyst Projekt zu verknüpfen

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Führen Sie einen der folgenden Schritte aus, um die installierte Jira Software-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wähle Einstellungen und dann Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie in Jira Software die Option Konfigurieren aus.
5. Wählen Sie auf der Registerkarte Verknüpfte Jira-Projekte die Option Jira-Projekt verknüpfen aus.
6. Wählen Sie im Dropdownmenü Jira-Site die Jira-Site aus, die das Projekt enthält, das Sie verknüpfen möchten.

7. Wählen Sie im Dropdownmenü für das Jira-Projekt das Projekt aus, das Sie mit einem Projekt verknüpfen möchten. CodeCatalyst
8. Wählen Sie im CodeCatalyst Drop-down-Menü für das CodeCatalyst Projekt das Projekt aus, das Sie mit einem Jira-Projekt verknüpfen möchten.
9. Wählen Sie Verknüpfen.

Sobald ein Jira-Projekt mit einem CodeCatalyst Projekt verknüpft ist, ist der Zugriff auf CodeCatalyst Issues vollständig deaktiviert und Issues im CodeCatalyst Navigationsbereich wird durch ein Jira-Issues-Element ersetzt, das auf das Jira-Projekt verweist.

## Die Verknüpfung eines Jira-Projekts mit einem Projekt aufheben CodeCatalyst

Durch das Aufheben der Verknüpfung eines Projekts wird das Jira-Projekt, einschließlich Planungselementen oder Entwicklungsinformationen, nicht gelöscht und es werden auch keine Änderungen daran vorgenommen. Sobald Sie die Verknüpfung mit einem Jira-Projekt aufheben, stehen die Jira-Probleme des Projekts jedoch nicht mehr für eine Verknüpfung mit dem Projekt zur Verfügung. CodeCatalyst

Um die Verknüpfung eines Jira-Projekts aufzuheben

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navigiere zu deinem CodeCatalyst Bereich.
3. Führen Sie einen der folgenden Schritte aus, um die installierte Jira Software-Erweiterung für Ihren Bereich anzuzeigen:
  - a. Wähle Einstellungen und dann Installierte Erweiterungen aus.
  - b. Wählen Sie  im oberen Menü das Katalogsymbol.
4. Wählen Sie auf der Registerkarte Verknüpfte Jira-Projekte das Jira-Projekt aus, dessen Verknüpfung Sie aufheben möchten.
5. Wählen Sie Jira-Projekt trennen aus.
6. Prüfen Sie im Dialogfeld „Verknüpfung aufheben“, welche Auswirkungen das Aufheben der Verknüpfung mit dem Repository hat.
7. Geben Sie unlink in das Texteingabefeld ein und wählen Sie Unlink aus.

## Ein Jira-Issue mit einer CodeCatalyst Pull-Anfrage verknüpfen

Du kannst Pull-Requests, die in einem CodeCatalyst Quell-Repository erstellt wurden, mit Jira-Issues verknüpfen. Nach dem Verknüpfen eines Jira-Issues wird das Issue als Eigenschaft des Pull-Requests angezeigt. Infolgedessen werden Pull-Request-Ereignisse, Workflow-Ereignisse und Bereitstellungereignisse an Jira gesendet und dem Jira-Issue hinzugefügt. Pull-Requests können mit einem oder mehreren Jira-Issues verknüpft werden. Du kannst nur Pull-Requests verknüpfen, die sich in einem CodeCatalyst Quell-Repository befinden, nicht solche, die sich in einem Drittanbieter-Repository befinden. GitHub Bevor du Jira-Issues mit einem Pull Request verknüpfen kannst, muss dein Jira-Projekt mit dem CodeCatalyst Projekt verknüpft werden. Weitere Informationen zum Verknüpfen eines Jira-Projekts mit einem CodeCatalyst Projekt finden Sie unter [Verwaltung von Jira-Seiten in CodeCatalyst](#)

### Note

Sie können keine Pull-Anfrage ohne ein Quell-Repository mit zwei Branches in Ihrem CodeCatalyst Projekt erstellen. Weitere Informationen zu Pull Requests findest du unter [Arbeiten mit Pull Requests in CodeCatalyst](#).

Um Jira-Issues mit einem CodeCatalyst Pull Request zu verknüpfen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem CodeCatalyst Projekt.
3. Wählen Sie im Navigationsbereich Code und dann Pull Requests aus.
4. Wählen Sie Pull-Request erstellen aus, um die Pull-Request-Details einzugeben.
5. Wählen Sie im Drop-down-Menü Quell-Repository das Quell-Repository aus, mit dem Sie einen Pull-Request verknüpfen möchten.
6. Wählen Sie im Drop-down-Menü Quell-Branch den Branch aus, der die Änderungen enthält, die Sie überprüfen möchten.
7. Wählen Sie im Dropdownmenü Zielzweig den Zweig aus, in dem Sie die überprüften Änderungen zusammenführen möchten.
8. Geben Sie im Texteingabefeld Titel der Pull-Anfrage den Titel Ihrer Pull-Requests ein.
9. Wähle „Probleme verknüpfen“ für die Jira-Probleme — optionales Feld aus, wähle das Drop-down-Menü aus und suche nach den Jira-Vorgängen, die du aus dem verknüpften Jira-Projekt hinzufügen möchtest.

10. Wählen Sie die Jira-Issues aus, die Sie der Pull-Anfrage hinzufügen möchten.
11. Wähle Create, um den Pull Request zu erstellen.

Sobald du Jira-Issues mit einer CodeCatalyst Pull-Anfrage verknüpft hast, ist eine Zusammenfassung der Pull-Anfrage verfügbar. Die Zusammenfassung umfasst Workflow-Ausführungen, verknüpfte Probleme, erforderliche Prüfer, optionale Prüfer und den Autor.

#### Note

Die mit dem Jira-Problem verknüpften Informationen „Beauftragter“ und „Erstellt von“ sind in nicht verfügbar. CodeCatalyst

Nach dem Verknüpfen einer Pull-Anfrage ermöglichen das synchronisierte CodeCatalyst Projekt und das Jira-Projekt, dass Aktualisierungen von CodeCatalyst in Ihrem Jira-Projekt berücksichtigt werden. Der Status des verknüpften Pull-Requests und alle Workflow-Ereignisse im Zusammenhang mit dem Pull Request werden in der Jira-Ausgabe angezeigt, wenn sie in Jira angezeigt wird. Weitere Informationen zum Anzeigen von CodeCatalyst Ereignissen in Jira finden Sie unter. [CodeCatalyst Ereignisse in Jira anzeigen](#)

## CodeCatalyst Ereignisse in Jira anzeigen

Wenn deine CodeCatalyst Projekte und Jira-Projekte verknüpft sind, spiegeln sich der Zusammenfassungsstatus der Pull-Anfrage und der Status der zugehörigen CodeCatalyst Workflow-Ereignisse in deinem Jira-Problem wider. Wenn du beispielsweise eine Pull-Anfrage schließt oder zusammenführst CodeCatalyst, spiegelt sich die Statusaktualisierung im Jira-Issue wider. CodeCatalyst Workflow-CI/CD-Ereignisse, die sich auf eine CodeCatalyst Pull-Anfrage beziehen, werden synchronisiert, sodass eine erfolgreiche Workflow-Ausführung auch an das Jira-Problem gesendet wird.

Um CodeCatalyst Ereignisse in einem Jira-Problem anzuzeigen

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigieren Sie zu Ihrem CodeCatalyst Projekt.
3. Wählen Sie im CodeCatalyst Navigationsbereich Code, dann Pull Requests und dann den Pull Request mit dem Jira-Issue aus, den Sie in Ihrem Jira-Projekt anzeigen möchten.

4. Wählen Sie im Bereich Zusätzliche Informationen das Jira-Problem aus, das Sie in Ihrem Jira-Projekt anzeigen möchten.
5. Wählen Sie im Bereich Details des Jira-Projekts die Option Pull-Request aus, die unter Entwicklung aufgeführt ist, um Details der Pull-Anfrage zu sehen.
6. (Optional) Um die neuesten Builds zu sehen, wählen Sie den Tab Builds.
7. (Optional) Um den Entwicklungsstatus zu sehen, wählen Sie den Tab Deployments.

## Jira-Probleme suchen in CodeCatalyst

Nachdem Sie ein Jira-Projekt verknüpft haben, können Sie das verknüpfte Jira-Projekt mithilfe der CodeCatalyst globalen Suchleiste nach Problemen durchsuchen. Sie können auch in einer Pull-Anfrage nach Jira-Issues suchen CodeCatalyst und gleichzeitig auf die Issues verlinken. Weitere Informationen zum Verknüpfen von Jira-Issues mit einer CodeCatalyst Pull-Anfrage finden Sie unter [Ein Jira-Issue mit einer CodeCatalyst Pull-Anfrage verknüpfen](#)

So suchen Sie in verknüpften Jira-Projekten nach Jira-Issues

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Navigieren Sie zu Ihrem CodeCatalyst Projekt.
3. Suchen Sie in der globalen Suchleiste in einem verknüpften Jira-Projekt nach Issues oder Jira-Issues, die Sie mit einer Pull-Anfrage verknüpfen möchten.

# Suche in CodeCatalyst

Verwenden Sie die Suchleiste oder ein spezielles Fenster mit Suchergebnissen, CodeCatalyst um Code, Probleme, Projekte und Benutzer zu durchsuchen CodeCatalyst.

Sie können Ressourcen in Ihrem Bereich und in Ihren Projekten finden, indem Sie Abfragen wie Name, Beschreibung und Status in die Suchleiste eingeben. Sie können Ihre Suchanfragen auch mithilfe der Sprache der Suchanfrage verfeinern.

## Themen

- [Verfeinern Sie Ihre Suchabfrage](#)
- [Überlegungen bei der Arbeit mit der Suche](#)
- [Referenz zu durchsuchbaren Feldern](#)

## Um zu suchen

1. Geben Sie in der Suchleiste in der oberen Navigationsleiste eine Suchabfrage ein.
2. (Optional) Verfeinern Sie Ihre Suchabfrage mithilfe CodeCatalyst der Sprache der Suchanfrage. Weitere Informationen finden Sie unter [Verfeinern Sie Ihre Suchabfrage](#).
3. Führen Sie eine der folgenden Aktionen aus:
  - Um innerhalb des Projekts, in dem Sie sich gerade befinden, nach Ressourcen zu suchen, wählen Sie Dieses Projekt aus.
  - Um in allen Projekten in dem Bereich, in dem Sie sich gerade befinden, nach Ressourcen zu suchen, wählen Sie Dieser Bereich.
4. Zeigen Sie Suchergebnisse in einem speziellen Suchergebnisfenster an, indem Sie einen der folgenden Schritte ausführen:
  - Wählen Sie unten im Fenster mit den Schnellsuchergebnissen die Option Alle Ergebnisse in Projektname | Space-Name anzeigen, um alle Suchergebnisse anzuzeigen.
  - Drücken Sie die Eingabetaste, um alle Suchergebnisse anzuzeigen.



 Tip

Erwähnen Sie andere Projektbenutzer in einem Pull-Request-Kommentar oder einer Beschreibung oder in einem Problemkommentar oder einer Problembeschreibung, indem Sie das @-Zeichen gefolgt von ihrem Anzeigenamen oder Benutzernamen verwenden. Du kannst auch Links zu Ressourcen wie Problemen oder Codedateien erstellen, indem du das @-Zeichen gefolgt vom Namen des Problems oder der Codedatei verwendest.

## Verfeinern Sie Ihre Suchabfrage

Wenn Sie nach der Suche nicht finden, wonach Sie suchen, können Sie Ihre Suche mit CodeCatalyst der speziellen Abfragesprache verfeinern. Einzelne Felder haben kein Zeichenlimit, aber die gesamte Abfrage hat ein Limit von 1.024 Zeichen.

Themen

- [Verfeinerung nach Typ](#)
- [Verfeinerung nach Feldern](#)
- [Verfeinerung mit booleschen Operatoren](#)
- [Verfeinerung nach Projekten](#)

## Verfeinerung nach Typ

Um den Umfang Ihrer Suche auf einen bestimmten Informationstyp zu beschränken, schließen Sie *type:result-type* in Ihre Suche ein, wobei der *Ergebnistyp* `code`, `issue`, `project` oder `user` ist.

Beispiele:

- `type:code AND java`— Zeigt Codeergebnisse in codebezogenen Feldern an, die „Java“ enthalten.

Weitere Informationen finden Sie unter [Felder codieren](#).

- `type:issue AND Bug`— Zeigt Problemergebnisse in problembezogenen Feldern an, die „Bug“ enthalten.

Weitere Informationen finden Sie unter [Problemfelder](#).

- `type:user AND MaryMajor`— Zeigt Benutzerergebnisse in benutzerbezogenen Feldern an, die „“ enthalten. `MaryMajor`

Weitere Informationen finden Sie unter [Benutzerfelder](#).

- `type:project AND Datafeeder`— Zeigt Projektergebnisse an, die „Datafeeder“ enthalten.

Weitere Informationen finden Sie unter [Bereiche des Projekts](#).

## Verfeinerung nach Feldern

Um den Suchbereich auf ein bestimmtes Feld zu beschränken, schließen *field-name:query* Sie die Suche mit ein, wobei *Fieldname* `title,username,projectdescription`, usw. ist und *Abfrage* der Text ist, nach dem Sie suchen. Eine Liste der Felder finden Sie unter [Referenz zu durchsuchbaren Feldern](#) Sie können mithilfe von Klammern nach mehreren Abfragen suchen.

Beispiele:

- `title:bug`— Zeigt Ergebnisse an, bei denen der Titel „Bug“ enthält.
- `username:John`— Zeigt Ergebnisse an, bei denen der Benutzername „John“ enthält.
- `project:DataFeeder`— Ergebnisse im Projekt „DataFeeder“ anzeigen. Bei der Abfrage wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- `description:overview`— Zeigt Ergebnisse an, bei denen die Beschreibung „Überblick“ enthält.

## Verfeinerung mit booleschen Operatoren

Um Einschränkungen für Suchausdrücke festzulegen, können Sie die booleschen Operatoren, und verwenden. AND OR NOT Wenn Sie mehrere Phrasen auflisten, werden CodeCatalyst sie OR standardmäßig mit verknüpft. Sie können Suchausdrücke mithilfe von Klammern gruppieren.

- `exception AND type:code`— Nur Codeergebnisse für „Ausnahme“ anzeigen.
- `path:README.md AND repo:ServerlessAPI`— Zeigt Ergebnisse für Pfade mit „README.md“ an, in denen das Repository „ServerlessAPI“ heißt.
- `buildspec.yml AND (repo:ServerlessAPI OR ServerlessWebApp)`— Zeigt Ergebnisse für „buildspec.yml“ an, wobei das Repository „ServerlessAPI“ oder „“ ist. `ServerlessWebApp`
- `path:java NOT (path:py OR path:ts)`— Zeigt Ergebnisse an, bei denen der Pfad „java“, aber nicht „py“ oder „ts“ enthält.

## Verfeinerung nach Projekten

Um den Umfang Ihrer Suche auf ein bestimmtes Projekt zu beschränken, schließen Sie Ihre Suche mit ein, wobei *Name* für das Projekt steht, *project:name AND query* in dem Sie suchen, und *Abfrage* für den Inhalt, nach dem Sie suchen.

- `project:name AND query`— Zeigt Ergebnisse an, bei denen der Pfad die Abfrage und den Projektnamen enthält.

## Überlegungen bei der Arbeit mit der Suche

Verzögerte Inhaltsaktualisierungen — Es kann mehrere Minuten dauern, bis Inhaltsaktualisierungen, wie z. B. Namensänderungen oder Neuzuweisungen von Problemen, in den Suchergebnissen angezeigt werden. Umfangreiche Updates, wie z. B. eine Codebasis-Migration, können länger dauern, bis sie in den Suchergebnissen angezeigt werden.

Sonderzeichen maskieren — Die folgenden Sonderzeichen müssen bei Ihren Suchanfragen besonders berücksichtigt werden: `+ - & & || ! ( ) { } [ ] ^ " ~ * ? : \`. Sonderzeichen haben keinen Einfluss auf die Abfrage, und Sie müssen sie entweder entfernen oder sie maskieren. Um ein Zeichen zu maskieren, fügen Sie vor dem Zeichen einen umgekehrten Schrägstrich (`\`) hinzu. Beispielsweise sollte die Suchabfrage `[Feature]` entweder `Feature` oder `\ [Feature\]` lauten.

Eingrenkte Suche — Bei der Suche wird nicht zwischen Groß- und Kleinschreibung unterschieden. Wenn Sie ausschließlich in Kleinbuchstaben suchen, wird verhindert, dass Ihre Abfragen Wörter bei der Änderung der Groß-/Kleinschreibung aufteilen. Wenn Sie beispielsweise nur nach `MyService` und suchen möchten, sollten Sie eine Abfrage in Erwägung ziehen `MyService`, um Ergebnisse `myservice` zu vermeiden, die nur oder enthalten. `my service`

Bei der Suche werden Wörter und Wortteile standardmäßig mit einer ODER-Verknüpfung verknüpft. `new function` könnte beispielsweise Ergebnisse zurückgeben, die `new function` sowohl als auch nur `new` oder `function` enthalten. Um Letzteres zu vermeiden, kombinieren Sie mehrere Wörter mit `AND`. Sie können beispielsweise suchen `new AND function`.

Standard-Branches — Die Suche gibt nur Codeergebnisse aus dem letzten Commit im Standard-Branche eines Quell-Repositorys zurück. Um Code in anderen Branches oder Commits zu finden, erwägen Sie, [das Repository lokal zu klonen](#), [den Branch in einer Entwicklungsumgebung zu öffnen](#) oder [die Branches und Details in der CodeCatalyst Benutzeroberfläche anzusehen](#). Eine Änderung

des Standard-Branches führt zu Aktualisierungen der Dateien, die über die Suche auffindbar sind. Weitere Informationen finden Sie unter [Den Standardzweig für ein Repository anzeigen und ändern](#).

## Referenz zu durchsuchbaren Feldern

CodeCatalyst durchsucht die folgenden Felder, wenn Sie Suchanfragen eingeben. Aliase sind ein weiterer Name, mit dem Sie in der erweiterten Abfragesprache auf das Feld verweisen können.

### Felder codieren

| Feld             | Alias  | Beschreibung   |
|------------------|--------|--|
| Name der Filiale | branch | Name des Zweigs, in dem sich die Codedatei befindet.   |
| Code             | –      | Informationen über den Codeinhalt in Form von Codefragmenten, die Teile des Quellcodes angeben, die der Suche entsprachen.   |
| CommitID         | –      | Commit-ID des Commits, in dem die zurückgegebene Codedatei zuletzt aktualisiert wurde. Kann die Commit-ID an der Spitze des unter angegebenen Branchnamens sein, muss aber nicht.<br>branchName          |
| CommitMessage    | –      | Commit-Nachricht des Commits, in dem die Codedatei zuletzt aktualisiert wurde. Kann die Commit-Nachricht an der Spitze des unter angegebenen Branchnamens sein oder auch nicht.<br>branchName Wenn keine |

| Feld            | Alias       | Beschreibung   |
|-----------------|-------------|--|
|                 |             | Commit-Nachricht angegeben wurde, ist dieser Wert eine leere Zeichenfolge.   |
| filePath        | path (Pfad) | Dateipfad dieser Codedatei.  |
| lastUpdatedBy   | –           | CodeCatalyst Benutzer, der die Codedatei zuletzt aktualisiert hat. Wenn der Benutzername nicht verfügbar ist, ist dieser Wert die E-Mail-Adresse des Benutzers, wie sie in der Git-Konfigurationsdatei konfiguriert ist. |
| lastUpdatedByID | –           | Vom System generierte eindeutige ID des Benutzers , der die Codedatei zuletzt aktualisiert hat. Wenn die Benutzer-ID nicht verfügbar ist, kann es sich bei diesem Wert um die E-Mail-Adresse des Benutzers handeln.      |
| lastUpdatedTime | –           | Uhrzeit, zu der die Suchdaten zuletzt mit dem Commit aktualisiert wurden, der die Codedatei enthielt (im UTC-Zeitstempel (Coordinated Universal Time)).  |
| projectId       | –           | Vom System generierte eindeutige ID des Projekts.  |

| Feld                 | Alias                 | Beschreibung   |
|----------------------|-----------------------|--|
| projectName          | Projektnamen, Projekt | Zeigt den Namen des Projekts an, das das Quell-Repository enthält, in das die Codedatei übergeben wurde. |
| Repository-ID        | Repository-ID         | Vom System generierte eindeutige ID des Quell-Repositorys.   |
| Name des Repositorys | Repository, Repo      | Zeigt den Namen des Quell-Repositorys an, in das die Codedatei übertragen wurde.                         |

## Problemfelder

| Feld                          | Alias                   | Beschreibung  |
|-------------------------------|-------------------------|---|
| IDs von zugewiesenen Personen | ID des Bevollmächtigten | Vom System generierte eindeutige IDs der Benutzer, die dem Problem zugewiesen wurden. |
| Beauftragte                   | Rechtsnachfolger(in)    | Benutzernamen der Benutzer, die dem Problem zugewiesen wurden.                        |
| Erstellt von                  | –                       | Anzeigenname des Benutzers, der das Problem erstellt hat.                             |
| createdById                   | –                       | Vom System generierte eindeutige ID des Benutzers, der das Problem erstellt hat.      |
| Erstellungszeit               | –                       | Uhrzeit, zu der das Problem erstellt wurde (im UTC-Zeits                              |

| Feld            | Alias    | Beschreibung   |
|-----------------|----------|--|
|                 |          | tempel (Coordinated Universal Time)).  |
| description     | –        | Beschreibung des Problems.   |
| ist archiviert  | archived | Boolescher Wert, der angibt, ob das Problem in einem archivierten Zustand erstellt werden soll.    |
| Ist geblockt    | blocked  | Boolescher Wert, der angibt, ob das Problem als blockiert markiert ist.                            |
| LabelIDs        | Label-ID | Vom System generierte eindeutige IDs der Labels für ein Problem.                                   |
| lastUpdatedBy   | –        | Zeigt den Namen des Benutzers an, der das Problem zuletzt aktualisiert hat.                        |
| lastUpdatedByID | –        | Vom System generierte eindeutige ID des Benutzers , der das Problem zuletzt aktualisiert hat.      |
| lastUpdatedTime | –        | Uhrzeit der letzten Aktualisierung des Problems (im UTC-Zeitstempel (Coordinated Universal Time)). |
| priority        | –        | Priorität des Problems, falls eine zugewiesen wurde.   |

| Feld        | Alias                 | Beschreibung   |
|-------------|-----------------------|--|
| projectId   | –                     | Vom System generierte eindeutige ID des Projekts.  |
| projectName | Projektnamen, Projekt | Projekt, in dem dieses Problem gefunden werden kann.   |
| ShortID     | –                     | Verkürzte, automatisch inkrementierende Kennung für das Problem.                             |
| status      | –                     | Status des Problems, der angibt, ob das Problem im Backlog oder in einer Spalte an Bord ist. |
| Status-ID   | –                     | System-ID des Status.  |
| Titel       | –                     | Titel der Ausgabe.   |

## Bereiche des Projekts

| Feld            | Alias   | Beschreibung   |
|-----------------|---------|--|
| description     | –       | Beschreibung des Projekts.   |
| lastUpdatedTime | –       | Uhrzeit der letzten Aktualisierung der Projektmetadaten (im UTC-Zeitstempel (Coordinated Universal Time)). |
| projectName     | project | Name des Projekts im Bereich.  |
| Projektpfad     | –       | URL-routbarer Name des Projekts, der bei der Projekter   |



| Feld | Alias | Beschreibung  |
|------|-------|---|
|      |       | stellung definiert wurde. Wird in URLs verwendet, die den Projektnamen erfordern. |

## Benutzerfelder

| Feld            | Alias    | Beschreibung   |
|-----------------|----------|--|
| displayName     | –        | Für den Benutzer verwendet er Name in CodeCatalyst. Anzeigenamen sind nicht eindeutig.   |
| email           | –        | E-Mail-Adresse des Benutzers   |
| lastUpdatedTime | –        | Uhrzeit der letzten Aktualisierung der Benutzermetadaten (im UTC-Zeitstempel (Coordinated Universal Time)).  |
| userName        | username | Benutzername, den der Benutzer bei der Registrierung für CodeCatalyst ausgewählt hat. Im Gegensatz zu Anzeigenamen können Benutzernamen nicht geändert werden. |

# Problembhebung bei Amazon CodeCatalyst

Die folgenden Informationen können Ihnen helfen, häufig auftretende Probleme in zu beheben CodeCatalyst. Sie können auch den CodeCatalyst Amazon-Gesundheitsbericht verwenden, um festzustellen, ob es Serviceprobleme gibt, die sich auf Ihre Erfahrung auswirken könnten.

## Themen

- [Behebung allgemeiner Zugriffsprobleme](#)
- [Behebung von Support-Problemen](#)
- [Amazon CodeCatalyst ist teilweise oder vollständig nicht verfügbar](#)
- [Ich kann kein Projekt erstellen in CodeCatalyst](#)
- [Ich möchte Feedback einreichen in CodeCatalyst](#)
- [Behebung von Problemen mit Quell-Repositorys](#)
- [Problembhebung bei Projekten und Blueprints](#)
- [Behebung von Problemen mit Workflows](#)
- [Behebung von Problemen mit der Suche in CodeCatalyst](#)
- [Beheben von Problemen mit Konten, die Ihrem Bereich zugeordnet sind](#)
- [Behebung von Problemen mit Entwicklungsumgebungen](#)
- [Behebung von Problemen mit Problemen](#)
- [Behebung von Problemen zwischen Amazon CodeCatalyst und den AWS SDKs oder dem AWS CLI](#)

## Behebung allgemeiner Zugriffsprobleme

### Ich habe mein Passwort vergessen

Problem: Ich habe das Passwort vergessen, das ich für meine AWS Builder-ID und Amazon verwende CodeCatalyst.

Mögliche Lösungen: Dieses Problem lässt sich am einfachsten beheben, indem Sie Ihr Passwort zurücksetzen.

1. Öffnen Sie [Amazon CodeCatalyst](#) und geben Sie Ihre E-Mail-Adresse ein. Klicken Sie nun auf Continue (Weiter).

2. Wählen Sie Passwort vergessen?
3. Wir senden Ihnen eine E-Mail mit einem Link, über den Sie Ihr Passwort ändern können. Wenn Sie die E-Mail nicht in Ihrem Posteingang sehen, überprüfen Sie Ihren Spam-Ordner.

## Amazon CodeCatalyst ist teilweise oder vollständig nicht verfügbar

**Problem:** Ich habe zur CodeCatalyst Konsole navigiert oder bin einem Link gefolgt, aber es wird ein Fehler angezeigt.

**Mögliche Lösungen:** Die häufigsten Gründe für dieses Problem sind, dass Sie entweder einem Link zu einem Projekt oder einem Bereich gefolgt sind, zu dem Sie nicht eingeladen wurden, oder dass ein Problem mit der allgemeinen Verfügbarkeit des Dienstes vorliegt. Prüfen Sie im [Gesundheitsbericht](#), ob es bekannte Probleme mit dem Dienst gibt. Falls nicht, kontaktiere die Person, die dich zu dem Projekt oder Space eingeladen hat, und bitte um eine weitere Einladung. Wenn du zu keinem Projekt oder [Space eingeladen wurdest, kannst du dich registrieren und deinen eigenen Bereich und deine eigenen Projekte erstellen](#).

## Ich kann kein Projekt erstellen in CodeCatalyst

**Problem:** Ich möchte ein Projekt erstellen, aber die Schaltfläche Projekt erstellen wird als nicht verfügbar angezeigt, oder ich erhalte eine Fehlermeldung.

**Mögliche Lösungen:** Die häufigsten Gründe für dieses Problem sind, dass Sie bei der Konsole mit einer AWS Builder-ID angemeldet sind, die nicht über die Space-Administratorrolle verfügt. Sie benötigen diese Rolle, um Projekte in einem Space erstellen zu können.

Wenn Sie über diese Rolle verfügen und die Schaltfläche nicht als verfügbar angezeigt wird, liegt möglicherweise ein vorübergehendes Problem mit dem Dienst vor. Aktualisieren Sie Ihren Browser und versuchen Sie es erneut.

## Behebung von Support-Problemen

### Ich erhalte eine Fehlermeldung, wenn ich auf Amazon AWS Support zugreife CodeCatalyst

**Problem:** Wenn ich die CodeCatalyst Option AWS Support für Amazon wähle, erhalte ich die folgende Fehlermeldung:

## Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the AWS-Konto that is the billing account for the space.

Mögliche Lösungen: Fügen Sie die erforderliche Rolle dem AWS-Konto Rechnungskonto für den Bereich hinzu. Das als Rechnungskonto für den Bereich angegebene Konto verwendet die `AWSRoleForCodeCatalystSupport` Rolle und die `AmazonCodeCatalystSupportAccess` verwaltete Richtlinie. Weitere Informationen finden Sie unter [Erstellen der `AWSRoleForCodeCatalystSupport` Rolle für Ihr Konto und Ihren Bereich](#).

### Note

Eine AWS Builder-ID kann nur Unterstützung für den Alias erhalten, mit dem sie authentifiziert wurde, und nur für Ressourcen, die auf den entsprechenden Berechtigungen basieren. CodeCatalyst Support für Konten und Abrechnung ist für alle Benutzer in diesem Bereich verfügbar. Entwickler können jedoch nur Unterstützung für Ressourcen und Informationen erhalten, für die sie über entsprechende Berechtigungen verfügen CodeCatalyst.

## Ich kann keine Anfragen für technischen Support für meine Unterkunft erstellen

Problem: Ich kann keine Anfragen für technischen Support für meine Unterkunft erstellen.

Korrekturen: Dem Space-Abrechnungskonto muss ein Business Support- oder Enterprise Support-Plan hinzugefügt werden, damit Benutzer im Space Anfragen für technischen Support erstellen können. Bitten Sie Ihren Space-Administrator, Ihrem Space-Abrechnungskonto einen AWS Support Tarif hinzuzufügen, oder besuchen Sie <https://repost.aws/>, um die AWS Community zu fragen.

## Mein Konto für Supportanfragen ist nicht mehr mit meinem Bereich in verbunden CodeCatalyst

Problem: Mein Konto für Supportanfragen ist nicht mehr mit meinem Bereich in verbunden CodeCatalyst.

**Behebung:** Wenn ein Benutzer mit der Rolle „Space-Administrator“ das Space-Abrechnungskonto wechselt, werden der AWS Support Tarif und alle zugehörigen Fälle vom Space getrennt. Die mit dem alten Space-Abrechnungskonto verknüpften AWS Support Fälle sind in AWS Support Amazon nicht mehr sichtbar CodeCatalyst. Der Root-Benutzer für dieses Rechnungskonto kann alte Fälle über anzeigen und lösen AWS Management Console und IAM-Berechtigungen einrichten, damit andere Benutzer alte Fälle einsehen und lösen können. AWS Support Sie können über das nicht mehr technischen Support für CodeCatalyst das alte Space-Abrechnungskonto in Anspruch nehmen AWS Management Console, aber Sie können technischen Support für andere Dienste erhalten, bis Ihr AWS Support Plan gekündigt wird.

Weitere Informationen finden Sie im AWS Support Benutzerhandbuch unter [Aktualisierung, Lösung und erneutes Öffnen Ihres Falls](#).

## Ich kann keinen Support-Fall für ein anderes AWS-Service in AWS Support for Amazon eröffnen CodeCatalyst

**Problem:** Ich kann keinen Support-Fall für ein anderes AWS-Service Mitglied AWS Support eröffnen CodeCatalyst.

**Mögliche Lösungen:** Sie können CodeCatalyst Supportanfragen nur von AWS Support für aus öffnen CodeCatalyst. Wenn Sie Support für Dienste oder Ressourcen benötigen, die von CodeCatalyst einem anderen Service AWS, Amazon oder einem anderen Drittanbieter bereitgestellt werden, müssen Sie einen Fall über den AWS Management Console oder den Service-Support-Kanal eines Drittanbieters erstellen. Weitere Informationen finden Sie im AWS Support Benutzerhandbuch unter [Erstellen von Supportanfragen und Fallmanagement](#).

## Amazon CodeCatalyst ist teilweise oder vollständig nicht verfügbar

**Problem:** Ich habe zur CodeCatalyst Konsole navigiert oder bin einem Link gefolgt, aber es wird ein Fehler angezeigt.

**Mögliche Lösungen:** Die häufigsten Gründe für dieses Problem sind, dass Sie entweder einem Link zu einem Projekt oder einem Bereich gefolgt sind, zu dem Sie nicht eingeladen wurden, oder dass ein Problem mit der allgemeinen Verfügbarkeit des Dienstes vorliegt. Prüfen Sie im [Gesundheitsbericht](#), ob es bekannte Probleme mit dem Dienst gibt. Falls nicht, kontaktiere die Person, die dich zu dem Projekt oder Space eingeladen hat, und bitte um eine weitere Einladung. Wenn du zu keinem Projekt oder [Space eingeladen wurdest, kannst du dich registrieren und deinen eigenen Bereich und deine eigenen Projekte erstellen](#).

## Ich kann kein Projekt erstellen in CodeCatalyst

**Problem:** Ich möchte ein Projekt erstellen, aber die Schaltfläche Projekt erstellen wird als nicht verfügbar angezeigt, oder ich erhalte eine Fehlermeldung.

**Mögliche Lösungen:** Die häufigsten Gründe für dieses Problem sind, dass Sie bei der Konsole mit einer AWS Builder-ID angemeldet sind, die nicht über die Space-Administratorrolle verfügt. Sie benötigen diese Rolle, um Projekte in einem Space erstellen zu können.

Wenn Sie über diese Rolle verfügen und die Schaltfläche nicht als verfügbar angezeigt wird, liegt möglicherweise ein vorübergehendes Problem mit dem Dienst vor. Aktualisieren Sie Ihren Browser und versuchen Sie es erneut.

## Ich möchte Feedback einreichen in CodeCatalyst

**Problem:** Ich habe einen Fehler in gefunden CodeCatalyst und möchte Feedback einreichen.

**Mögliche Lösungen:** Sie können Feedback direkt in einreichen CodeCatalyst.

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie im Navigationsbereich Feedback geben aus.
3. Wählen Sie die Art des Feedbacks aus dem Drop-down-Menü aus und geben Sie Ihr Feedback ein.

## Behebung von Problemen mit Quell-Repositorys

Die folgenden Informationen können Ihnen helfen, häufig auftretende Probleme mit Quell-Repositorys in zu beheben. CodeCatalyst

### Themen

- [Ich habe die maximale Speicherkapazität für meinen Speicherplatz erreicht und erhalte Warnungen oder Fehler](#)
- [Ich erhalte eine Fehlermeldung, wenn ich versuche, ein CodeCatalyst Amazon-Quell-Repository zu klonen oder zu pushen](#)
- [Ich erhalte eine Fehlermeldung, wenn ich versuche, einen Commit oder einen Push in ein CodeCatalyst Amazon-Quell-Repository durchzuführen](#)
- [Ich benötige ein Quell-Repository für mein Projekt](#)

- [Mein Quell-Repository ist brandneu, enthält aber einen Commit](#)
- [Ich möchte einen anderen Zweig als meinen Standardzweig](#)
- [Ich erhalte E-Mails über Aktivitäten in Pull-Requests](#)
- [Ich habe mein persönliches Zugriffstoken \(PAT\) vergessen](#)
- [Eine Pull-Anfrage zeigt nicht die Änderungen an, die ich erwarte](#)
- [Ein Pull-Request zeigt den Status Nicht zusammenführbar](#)

## Ich habe die maximale Speicherkapazität für meinen Speicherplatz erreicht und erhalte Warnungen oder Fehler

Problem: Ich möchte Code in ein oder mehrere Quell-Repositorys übertragen CodeCatalyst, aber es wird ein Fehler angezeigt. In der Konsole wird auf der Quell-Repository-Seite eine Meldung angezeigt, dass ich das Speicherlimit für den Speicherplatz erreicht habe.

Mögliche Lösungen: Abhängig von Ihrer Rolle im Projekt oder Space können Sie entweder die Größe eines oder mehrerer Ihrer Quell-Repositorys reduzieren, ungenutzte Quell-Repositorys löschen oder Ihre Abrechnungsstufe auf eine Stufe mit mehr Speicherplatz ändern.

- Um die Größe eines Quell-Repositorys in einem Projekt zu reduzieren, können Sie ungenutzte Branches löschen. Weitere Informationen finden Sie unter [Um einen Zweig zu löschen \(Konsole\)](#) und [Rolle „Mitwirkender“](#).
- Um den Gesamtspeicher für einen Bereich zu reduzieren, können Sie ungenutzte Quell-Repositorys löschen. Weitere Informationen finden Sie unter [Löschen eines Quell-Repositorys](#) und [Rolle als Projektadministrator](#).
- Um den für Ihren Speicherplatz verfügbaren Speicherplatz zu erhöhen, können Sie Ihre Abrechnungsstufe auf eine Stufe mit mehr Speicherplatz ändern. Weitere Informationen finden Sie unter [Ändern Ihrer CodeCatalyst Abrechnungsstufe](#) im CodeCatalyst Amazon-Administratorhandbuch.

## Ich erhalte eine Fehlermeldung, wenn ich versuche, ein CodeCatalyst Amazon-Quell-Repository zu klonen oder zu pushen

Problem: Wenn ich versuche, ein Quell-Repository auf einen lokalen Computer oder in eine integrierte Entwicklungsumgebung (IDE) zu klonen, erhalte ich einen Berechtigungsfehler.

Mögliche Lösungen: Möglicherweise haben Sie kein Personal Access Token (PAT) für Ihre AWS Builder-ID, Sie haben Ihr Anmeldeinformationsverwaltungssystem möglicherweise nicht mit Ihrem PAT konfiguriert, oder Ihr PAT ist möglicherweise abgelaufen. Probieren Sie eine oder mehrere der folgenden Lösungen aus:

- Erstellen Sie ein persönliches Zugriffstoken (PAT). Weitere Informationen finden Sie unter [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#).
- Stellen Sie sicher, dass Sie eine Einladung zu dem Projekt angenommen haben, das das Quell-Repository enthält, und dass Sie immer noch Mitglied dieses Projekts sind. Sie können ein Quell-Repository nicht klonen, wenn Sie kein aktives Mitglied dieses Projekts sind. Melden Sie sich bei der Konsole an und versuchen Sie, zu dem Bereich und dem Projekt zu navigieren, in dem Sie ein Quell-Repository klonen möchten. Wenn Sie das Projekt nicht in der Projektliste für den Bereich sehen können, sind Sie entweder kein Mitglied dieses Projekts oder Sie haben eine Einladung zu diesem Projekt nicht angenommen. Weitere Informationen finden Sie unter [Eine Einladung annehmen und Ihre AWS Builder-ID erstellen](#).
- Vergewissern Sie sich, dass Ihr Clone-Befehl korrekt formatiert ist und Ihre AWS Builder-ID enthält. Beispielsweise:

```
https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

- Verwenden Sie den, AWS CLI um sicherzustellen, dass Ihrer AWS Builder-ID eine PAT zugeordnet ist und dass diese nicht abgelaufen ist. Wenn Sie noch keine haben oder die PAT abgelaufen ist, erstellen Sie eine. Weitere Informationen finden Sie unter [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#).
- Versuchen Sie, eine Entwicklungsumgebung zu erstellen, um mit dem Code im Quell-Repository zu arbeiten, anstatt ihn in ein lokales Repo oder eine IDE zu klonen. Weitere Informationen finden Sie unter [Erstellen einer Entwicklungsumgebung](#).

## Ich erhalte eine Fehlermeldung, wenn ich versuche, einen Commit oder einen Push in ein CodeCatalyst Amazon-Quell-Repository durchzuführen

Problem: Wenn ich versuche, in ein Quell-Repository zu pushen, erhalte ich einen Berechtigungsfehler.

Mögliche Lösungen: Möglicherweise haben Sie keine Rolle in dem Projekt, die es Ihnen ermöglicht, Codeänderungen an das Projekt zu übertragen und per Push zu übertragen. Sehen Sie sich Ihre



Rolle in dem Projekt an, in dem Sie versuchen, Änderungen in ein Quell-Repository zu übertragen. Weitere Informationen finden Sie unter [Mitglieder in einem Projekt anzeigen](#) und [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

Wenn Sie über eine Rolle verfügen, die das Commit und das Pushen von Änderungen ermöglicht, ist für den Branch, in dem Sie versuchen, Änderungen zu übernehmen, möglicherweise eine Branch-Regel konfiguriert, die Sie daran hindert, Codeänderungen in diesen Branch zu übertragen. Versuchen Sie stattdessen, einen Branch zu erstellen und Ihren Code in diesen Branch zu übertragen. Weitere Informationen finden Sie unter [Zulässige Aktionen für einen Branch mit Branch-Regeln verwalten](#).

## Ich benötige ein Quell-Repository für mein Projekt

**Problem:** Mein Projekt hat entweder kein Quell-Repository oder ich benötige ein anderes Quell-Repository für mein Projekt.

**Mögliche Lösungen:** Einige Projekte werden ohne Ressourcen erstellt. Wenn Sie Mitglied des Projekts sind, können Sie Quell-Repositorys für dieses Projekt in CodeCatalyst erstellen. Wenn jemand mit der Space-Administratorrolle die GitHub Repositorys installiert und sie mit einem GitHub Konto verbindet, können Sie eine Verknüpfung zu verfügbaren GitHub Repositorys herstellen, um sie Ihrem Projekt hinzuzufügen, sofern Sie die Projektadministratorrolle haben. Weitere Informationen finden Sie unter [Erstellen eines Quell-Repositorys und Verknüpfen eines Quell-Repositorys](#).

## Mein Quell-Repository ist brandneu, enthält aber einen Commit

**Problem:** Ich habe gerade ein Quell-Repository erstellt. Es sollte leer sein, aber es enthält einen Commit, einen Branch und eine README .md Datei.

**Mögliche Korrekturen:** Dies ist das erwartete Verhalten. Alle Quell-Repositorys CodeCatalyst enthalten einen ersten Commit, der den Standard-Branch auf `main` setzt und entweder Beispielcode (falls das Repository für ein Projekt mit einem Blueprint erstellt wurde, der Beispielcode enthielt) oder eine Markdown-Template-Datei für eine Readme-Datei des Repositorys enthält. Sie können zusätzliche Branches in der Konsole und in Git-Clients erstellen. Sie können Dateien in der Konsole erstellen und bearbeiten und Dateien in Entwicklungsumgebungen und Git-Clients löschen.

## Ich möchte einen anderen Zweig als meinen Standardzweig

**Problem:** Mein Quell-Repository hatte einen Standard-Branch mit dem Namen `main`, aber ich möchte einen anderen Branch als Standard-Branch.

Mögliche Lösungen: Sie können den Standard-Branch in den Quell-Repositorys in CodeCatalyst nicht ändern oder löschen. Sie können zusätzliche Branches erstellen und diese Branches in Quellaktionen in Workflows verwenden. Sie können sich auch dafür entscheiden, GitHub Repositorys zu verknüpfen und sie als Repositorys für Ihr Projekt zu verwenden.

## Ich erhalte E-Mails über Aktivitäten in Pull-Requests

Problem: Ich habe mich nicht registriert oder E-Mail-Benachrichtigungen über Pull-Request-Aktivitäten konfiguriert, erhalte sie aber trotzdem.

Mögliche Korrekturen: E-Mail-Benachrichtigungen werden automatisch über Pull-Request-Aktivitäten gesendet. Weitere Informationen finden Sie unter [Mit Pull-Requests in Amazon arbeiten CodeCatalyst](#).

## Ich habe mein persönliches Zugriffstoken (PAT) vergessen

Problem: Ich habe ein PAT zum Klonen, Übertragen und Abrufen von Code für Quell-Repositorys verwendet, aber ich habe den Wert für mein Token verloren und kann es in der CodeCatalyst Konsole nicht finden.

Mögliche Lösungen: Dieses Problem lässt sich am schnellsten lösen, indem Sie ein weiteres PAT erstellen und Ihren Credential Manager oder Ihre IDE so konfigurieren, dass dieses neue PAT verwendet wird. Wir zeigen den Wert einer PAT nur an, wenn Sie sie erstellen. Wenn Sie diesen Wert verlieren, kann er nicht abgerufen werden. Weitere Informationen finden Sie unter [Verwaltung persönlicher Zugriffstoken in Amazon CodeCatalyst](#).

## Eine Pull-Anfrage zeigt nicht die Änderungen an, die ich erwarte

Problem: Ich habe eine Pull-Anfrage erstellt, sehe aber nicht die Änderungen, die ich erwarte, zwischen dem Quell- und dem Ziel-Branch.

Mögliche Korrekturen: Dies könnte durch eine Reihe von Problemen verursacht werden. Probieren Sie eine oder mehrere der folgenden Lösungen aus:

- Möglicherweise überprüfen Sie die Änderungen zwischen älteren Versionen, oder Sie sehen sich möglicherweise nicht die neuesten Änderungen an. Aktualisieren Sie Ihren Browser und stellen Sie sicher, dass Sie den Vergleich zwischen den Versionen ausgewählt haben, den Sie sich ansehen möchten.
- Nicht alle Änderungen in einer Pull-Anfrage können in der Konsole angezeigt werden. Sie können beispielsweise keine Git-Submodule in der Konsole anzeigen, sodass Sie auch keine

Unterschiede in einem Submodul in einer Pull-Anfrage anzeigen können. Einige Unterschiede sind möglicherweise zu groß, um angezeigt zu werden. Weitere Informationen finden Sie unter [Kontingente für Quell-Repositorys in CodeCatalyst](#) und [Eine Datei ansehen](#).

- In Pull-Requests werden die Unterschiede zwischen der Merge-Basis und der von Ihnen ausgewählten Revision angezeigt. Wenn du einen Pull Request erstellst, ist der Unterschied, der dir angezeigt wird, der Unterschied zwischen der Spitze des Quell-Branche und der Spitze des Ziel-Branche. Sobald der Pull Request erstellt wurde, besteht der angezeigte Unterschied zwischen der Revision und der zugehörigen Merge-Basis. Die Merge-Base ist der Commit, der bei der Erstellung der Revision die Spitze des Ziel-Branche war. Die Merge-Basis kann sich zwischen den Revisionen ändern. Weitere Informationen zu Unterschieden und Merge-Bases in Git finden Sie [git-merge-base](#) in der Git-Dokumentation.

## Ein Pull-Request zeigt den Status Nicht zusammenführbar

Problem: Ich möchte eine Pull-Anfrage zusammenführen, aber ihr Status wird als Nicht zusammenführbar angezeigt.

Mögliche Lösungen: Dies kann durch ein oder mehrere Probleme verursacht werden:

- Alle für deinen Pull Request erforderlichen Reviewer müssen einen Pull Request genehmigen, bevor er zusammengeführt werden kann. Sieh dir in der Liste der erforderlichen Prüfer alle Prüfer an, neben deren Namen ein Uhrensymbol angezeigt wird. Ein Uhrensymbol weist darauf hin, dass der Prüfer den Pull-Request nicht genehmigt hat.

### Note

Wenn ein erforderlicher Prüfer vor der Genehmigung des Pull Requests aus deinem Projekt entfernt wurde, kannst du den Pull Request nicht zusammenführen. Schließen Sie den Pull Request und erstellen Sie einen neuen Pull Request.

- Möglicherweise besteht ein Mergekonflikt zwischen dem Quell-Branche und dem Ziel-Branche. CodeCatalyst unterstützt nicht alle möglichen Git-Merge-Strategien und -Optionen. Sie können die Branches in einer Entwicklungsumgebung auf Merge-Konflikte untersuchen oder das Repository klonen und mithilfe von IDE- oder Git-Tools Merge-Konflikte finden und lösen. Weitere Informationen finden Sie unter [Eine Pull-Anfrage zusammenführen](#).

## Problembhebung bei Projekten und Blueprints

Dieser Abschnitt kann Ihnen helfen, einige häufig auftretende Probleme zu beheben, die bei der Arbeit mit Projekten und Blueprints in Amazon CodeCatalyst auftreten können.

### Java-API mit AWS Fargate Blueprint, fehlende Abhängigkeiten für Apache-Maven-3.8.6

**Problem:** Bei einem Projekt, das über die Java-API mit AWS Fargate Blueprint erstellt wurde, schlägt der Workflow fehl und es wird ein Fehler wegen fehlender Abhängigkeiten angezeigt. `apache-maven-3.8.6` Der Workflow schlägt fehl und es wird eine Ausgabe ähnlich dem folgenden Beispiel angezeigt:

```
Step 8/25 : RUN wget https://d1cdn.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://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz
[0m[91mResolving d1cdn.apache.org (d1cdn.apache.org)...
[0m[91m151.101.2.132, 2a04:4e42::644
Connecting to d1cdn.apache.org (d1cdn.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://d1cdn.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
```

**Lösung:** Aktualisieren Sie das Blueprint-Dockerfile mithilfe der folgenden Schritte.

1. Geben Sie in der Suchleiste ein, `apache-maven-3.8.6` um das Dockerfile in dem Projekt zu suchen, das mit der Java-API mit Blueprint erstellt wurde. AWS Fargate
2. Aktualisieren Sie das Dockerfile (`/static-assets/app/Dockerfile`), das `maven:3.9.0-amazoncorretto-11` als Basisimage verwendet werden soll, und entfernen Sie die Abhängigkeit vom Paket. `apache-maven-3.8.6`
3. (Empfohlen) Wir empfehlen außerdem, die Maven-Heap-Größe auf 6 GB zu aktualisieren.

Im Folgenden finden Sie ein Beispiel für eine 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"]
```

## Moderner dreistufiger Blueprint-Workflow für Webanwendungen OnPullRequest schlägt mit einem Berechtigungsfehler für Amazon fehl CodeGuru

Problem: Wenn ich versuche, einen Workflow für mein Projekt auszuführen, schlägt die Ausführung des Workflows fehl und die folgende Meldung wird angezeigt:

```
Failed at codeguru_codereview: The action failed during runtime. View the action's logs for more details.
```

Lösung: Eine mögliche Ursache für diesen Aktionsfehler könnte auf fehlende Berechtigungen in der IAM-Rollenrichtlinie zurückzuführen sein, sodass in Ihrer Version der Servicerolle, die von CodeCatalyst in the connected AWS-Konto verwendet wird, die erforderlichen Berechtigungen für die erfolgreiche Ausführung der Aktion `codeguru_codereview` fehlen. Um dieses Problem zu beheben, muss entweder die Servicerolle mit den erforderlichen Berechtigungen aktualisiert werden, oder Sie müssen die für den Workflow verwendete Servicerolle so ändern, dass sie über die erforderlichen Berechtigungen für Amazon CodeGuru und Amazon CodeGuru Reviewer verfügt. Suchen Sie anhand der folgenden Schritte nach Ihrer Rolle und aktualisieren Sie die Rollenrichtlinien-Berechtigungen, damit der Workflow erfolgreich ausgeführt werden kann.

**Note**

Diese Schritte gelten für die folgenden Workflows in CodeCatalyst:

- Der OnPullRequestWorkflow wurde für Projekte bereitgestellt, die mit dem Blueprint für moderne dreistufige Webanwendungen in erstellt wurden. CodeCatalyst
- Workflows, die zu Projekten CodeCatalyst mit Aktionen hinzugefügt wurden, die auf Amazon CodeGuru oder Amazon CodeGuru Reviewer zugreifen.

Jedes Projekt enthält Workflows mit Aktionen, die eine Rolle und Umgebung verwenden, die von dem AWS-Konto mit Ihrem Projekt verbundenen Benutzer bereitgestellt CodeCatalyst werden. Der Workflow mit den Aktionen und den ihnen zugewiesenen Richtlinien wird in Ihrem Quell-Repository im Verzeichnis `/.codecatalyst/workflows` gespeichert. Das Ändern des Workflow-YAML ist nicht erforderlich, es sei denn, Sie fügen dem vorhandenen Workflow eine neue Rollen-ID hinzu. Informationen zu YAML-Vorlagenelementen und Formatierungen finden Sie unter [Referenz zur Workflow-Definition](#)

Dies sind die allgemeinen Schritte, die Sie befolgen müssen, um Ihre Rollenrichtlinie zu bearbeiten und den YAML-Workflow zu überprüfen.

Um im Workflow-YAML auf Ihren Rollennamen zu verweisen und die Richtlinie zu aktualisieren

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich. Navigiere zu deinem Projekt.
3. Wählen Sie CI/CD und dann Workflows.
4. Wählen Sie den Workflow mit dem Titel. OnPullRequest Wählen Sie die Registerkarte Definition.
5. Notieren Sie sich im Workflow-YAML im `role`: Feld unter der Aktion `codeguru_codereview` den Rollennamen. Dies ist die Rolle mit der Richtlinie, die Sie in IAM ändern werden. Das folgende Beispiel zeigt den Rollennamen.

```

1
2 Name: OnPullRequest
3 SchemaVersion: "1.0"
4 Triggers:
5   - Type: PULLREQUEST
6   Branches:
7     - main
8 Events:
9   - OPEN
10  - REVISION
11 Actions:
12  codeguru_codereview:
13    Identifier: aws/build@v1
14    Inputs:
15      Sources:
16        - WorkflowSource
17      Variables:
18        - Name: AWS_DEFAULT_REGION
19          Value: us-west-2
20    Outputs:
21      Artifacts:
22        - Name: codereview
23          Files:
24            - ./code-guru/*
25 Configuration:
26   Steps:
27     - Run: curl -OL https://github.com/aws/aws-codeguru-c
28     - Run: unzip aws-codeguru-cli.zip
29     - Run: export PATH=$PATH:./aws-codeguru-cli/bin
30     - Run: aws-codeguru-cli --root-dir ./src --no-prompt -
31 Environment:
32   Name: development
33 Connections:
34   - Name: connection-11-30
35     Role: CodeCatalystPreviewDevelopmentAdministrator-j
36

```

## 6. Führen Sie eine der folgenden Aktionen aus:

- (Empfohlen) Aktualisieren Sie die mit Ihrem Projekt verbundene Servicerolle mit den erforderlichen Berechtigungen für Amazon CodeGuru und Amazon CodeGuru Reviewer. Der Rolle wird ein Name `CodeCatalystWorkflowDevelopmentRole-spaceName` mit einer angehängten eindeutigen Kennung zugewiesen. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#). Fahren Sie mit den nächsten Schritten fort, um die Richtlinie in IAM zu aktualisieren.

### Note

Sie benötigen AWS Administratorzugriff auf die AWS-Konto Rolle und Richtlinie.

- Ändern Sie die für den Workflow verwendete Servicerolle in eine, die über die erforderlichen Berechtigungen für Amazon CodeGuru und Amazon CodeGuru Reviewer verfügt, oder erstellen Sie eine neue Rolle mit den erforderlichen Berechtigungen.
7. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.

Suchen Sie in der IAM-Konsole nach der Rolle aus Schritt 5, z. B.  
CodeCatalystPreviewDevelopmentRole

8. Ändern Sie in der Rolle aus Schritt 5 die Berechtigungsrichtlinie so, dass sie die `codeguru:*` Berechtigungen `codeguru-reviewer:*` und einschließt. Nach dem Hinzufügen dieser Berechtigungen sollte die Berechtigungsrichtlinie etwa wie folgt aussehen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "apigateway:*",
        "ecr:*",
        "ecs:*",
        "ssm:*",
        "codedeploy:*",
        "s3:*",
        "iam:DeleteRole",
        "iam:UpdateRole",
        "iam:Get*",
        "iam:TagRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam:CreatePolicy",
        "iam>DeletePolicy",
        "iam:CreatePolicyVersion",
        "iam>DeletePolicyVersion",
        "iam:PutRolePermissionsBoundary",
        "iam>DeleteRolePermissionsBoundary",
        "sts:AssumeRole",
      ]
    }
  ]
}
```



```
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:DescribeRules",
        "elasticloadbalancing:ModifyRule",
        "cloudwatch:DescribeAlarms",
        "sns:Publish",
        "sns:ListTopics",
        "codeguru-reviewer:*",
        "codeguru:*"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
```

9. Nachdem Sie die Richtlinienkorrekturen vorgenommen haben, kehren Sie zu der Workflow-Ausführung zurück CodeCatalyst und starten Sie die Ausführung erneut.

## Sie suchen immer noch nach einer Lösung für Ihr Problem?

Sie können zu [Amazon](#) gehen CodeCatalyst oder ein [Support-Feedback-Formular](#) ausfüllen. Geben Sie im Abschnitt Informationen anfordern unter Wie können wir Ihnen helfen, an, dass Sie ein CodeCatalyst Amazon-Kunde sind. Geben Sie so viele Details wie möglich an, damit wir Ihr Problem am effizientesten lösen können.

## Behebung von Problemen mit Workflows

Lesen Sie die folgenden Abschnitte, um Probleme im Zusammenhang mit Workflows in Amazon zu beheben CodeCatalyst. Weitere Informationen zu Workflows finden Sie unter [???](#).

### Themen

- [Wie behebe ich die Meldung „Der Workflow ist inaktiv“?](#)
- [Wie behebe ich den Fehler „Workflow-Definition enthält keine Fehler“?](#)
- [Wie behebe ich die Fehler „Anmeldeinformationen konnten nicht gefunden werden“ und `ExpiredToken` „?“](#)
- [Wie behebe ich die Fehler „Es konnte keine Verbindung zum Server hergestellt werden“?](#)
- [Warum fehlen CodeDeploy Felder im Visual Editor?](#)

- [Wie behebe ich Fehler bei den IAM-Fähigkeiten?](#)
- [Wie behebe ich „npm install“ -Fehler?](#)
- [Warum haben mehrere Workflows denselben Namen?](#)
- [Kann ich meine Workflow-Definitionsdateien in einem anderen Ordner speichern?](#)
- [Wie füge ich meinem Workflow nacheinander Aktionen hinzu?](#)
- [Warum wird mein Workflow erfolgreich validiert, schlägt aber zur Laufzeit fehl?](#)
- [Die automatische Erkennung erkennt keine Berichte für meine Aktion](#)
- [Meine Aktion schlägt bei automatisch erkannten Berichten fehl, nachdem ich Erfolgskriterien konfiguriert habe](#)
- [Autodiscovery generiert Berichte, die ich nicht haben möchte](#)
- [Auto-Discovery generiert viele kleine Berichte für ein einzelnes Test-Framework](#)
- [Die unter CI/CD aufgeführten Workflows stimmen nicht mit denen im Quell-Repository überein](#)
- [Ich kann keine Workflows erstellen oder aktualisieren](#)

## Wie behebe ich die Meldung „Der Workflow ist inaktiv“?

Problem: In der CodeCatalyst Konsole wird unter CI/CD, Workflows, Ihr Workflow mit der folgenden Meldung angezeigt:

```
Workflow is inactive.
```

Diese Meldung weist darauf hin, dass die Workflow-Definitionsdatei einen Trigger enthält, der nicht für den Zweig gilt, in dem Sie sich gerade befinden. Ihre Workflow-Definitionsdatei könnte beispielsweise einen PUSH Trigger enthalten, der auf Ihren main Zweig verweist, aber Sie befinden sich in einem Feature-Branch. Da die Änderungen, die Sie an Ihrem Feature-Branch vornehmen, sich nicht auf Workflow-Läufe beziehen und auch keine Workflow-Läufe in diesem Zweig starten, wird CodeCatalyst der Workflow in dem Branch außer Betrieb genommen und markiert ihn als inactive.

Mögliche Lösungen:

Wenn Sie einen Workflow für Ihren Feature Branch starten möchten, können Sie wie folgt vorgehen:

- Entfernen Sie in Ihrem Feature Branch in der Workflow-Definitionsdatei die Branches Eigenschaft aus dem Triggers Abschnitt, sodass sie wie folgt aussieht:

```
Triggers:  
- Type: PUSH
```

Diese Konfiguration bewirkt, dass der Trigger bei einem Push in einen beliebigen Branch aktiviert wird, einschließlich Ihres Feature-Branche. Wenn der Trigger aktiviert ist, CodeCatalyst wird ein Workflow-Lauf mit der Workflow-Definitionsdatei und den Quelldateien in dem Zweig gestartet, in den Sie pushen.

- Entfernen Sie in Ihrem Feature-Branch in der Workflow-Definitionsdatei den Triggers Abschnitt und führen Sie den Workflow manuell aus.
- Ändern Sie in Ihrem Feature-Branch in der Workflow-Definitionsdatei den PUSH Abschnitt so, dass er auf Ihren Feature-Branch verweist und nicht auf einen anderen Branch (wie main zum Beispiel).

#### Important

Achten Sie darauf, diese Änderungen nicht zu übernehmen, wenn Sie nicht beabsichtigen, sie wieder in Ihrem main Branch zusammenzuführen.

Weitere Informationen zum Bearbeiten der Workflow-Definitionsdatei finden Sie unter [Um einen Workflow zu bearbeiten](#).

Weitere Informationen zu Auslösern finden Sie unter [Arbeiten mit Auslösern](#).

## Wie behebe ich den Fehler „Workflow-Definition enthält **keine** Fehler“?

Problem: Es wird eine der folgenden Fehlermeldungen angezeigt:

Fehler 1:

Auf der Seite CI/CD, Workflows, sehen Sie unter dem Namen Ihres Workflows:

```
Workflow definition has n errors
```

Fehler 2:

Wenn Sie einen Workflow bearbeiten, klicken Sie auf die Schaltfläche „Validieren“ und die folgende Meldung wird oben in der CodeCatalyst Konsole angezeigt:

The workflow definition has errors. Fix the errors and choose Validate to verify your changes.

Fehler 3:

Nachdem Sie zur Detailseite Ihres Workflows navigiert haben, wird im Feld Workflow-Definition der folgende Fehler angezeigt:

*n* errors

Mögliche Lösungen:

- Wählen Sie CI/CD, wählen Sie Workflows und wählen Sie den Namen des Workflows aus, in dem der Fehler aufgetreten ist. Wählen Sie im Feld Workflow-Definition oben den Link zu dem Fehler aus. Details zum Fehler werden unten auf der Seite angezeigt. Folgen Sie den Tipps zur Fehlerbehebung in der Fehlermeldung, um das Problem zu beheben.
- Stellen Sie sicher, dass es sich bei der Workflow-Definitionsdatei um eine YAML-Datei handelt.
- Stellen Sie sicher, dass die YAML-Eigenschaften in der Workflow-Definitionsdatei auf der richtigen Ebene verschachtelt sind. Informationen darüber, wie Eigenschaften in der Workflow-Definitionsdatei verschachtelt werden sollten, finden Sie in der [Referenz zur Workflow-Definition](#) oder in der Dokumentation Ihrer Aktion, die mit dem Link von verknüpft ist. [Hinzufügen einer Aktion](#)
- Stellen Sie sicher, dass Sternchen (\*) und andere Sonderzeichen korrekt maskiert werden. Um sie zu maskieren, fügen Sie einfache oder doppelte Anführungszeichen hinzu. Beispielsweise:

```
Outputs:  
  Artifacts:  
    - Name: myartifact  
      Files:  
        - "**/*"
```

Weitere Hinweise zu Sonderzeichen in der Workflow-Definitionsdatei finden Sie unter [Richtlinien und Konventionen zur Syntax](#).

- Stellen Sie sicher, dass die YAML-Eigenschaften in der Workflow-Definitionsdatei die richtige Groß-/Kleinschreibung verwenden. Weitere Informationen zu Regeln für Groß- und Kleinschreibung finden Sie unter [Richtlinien und Konventionen zur Syntax](#). Informationen zur korrekten Groß- und Kleinschreibung der einzelnen Eigenschaften finden Sie in der [Referenz zur Workflow-Definition](#) oder in der Dokumentation zu Ihrer Aktion, die mit dem Link von verknüpft ist [Hinzufügen einer Aktion](#).

- Vergewissern Sie sich, dass die `SchemaVersion` Eigenschaft in der Workflow-Definitionsdatei vorhanden und auf die richtige Version eingestellt ist. Weitere Informationen finden Sie unter [SchemaVersion](#).
- Stellen Sie sicher, dass der `Triggers` Abschnitt in der Workflow-Definitionsdatei alle erforderlichen Eigenschaften enthält. Um die erforderlichen Eigenschaften zu ermitteln, wählen Sie den Trigger im [Visual Editor](#) aus und suchen Sie nach Feldern, in denen Informationen fehlen, oder schauen Sie in der Referenzdokumentation zu Triggern unter nach [Triggers](#).
- Stellen Sie sicher, dass die `DependsOn` Eigenschaft in der Workflow-Definitionsdatei richtig konfiguriert ist und keine zirkulären Abhängigkeiten mit sich bringt. Weitere Informationen finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).
- Stellen Sie sicher, dass der `Actions` Abschnitt in der Workflow-Definitionsdatei mindestens eine Aktion enthält. Weitere Informationen finden Sie unter [Aktionen](#).
- Stellen Sie sicher, dass jede Aktion alle erforderlichen Eigenschaften enthält. Um die erforderlichen Eigenschaften zu ermitteln, wählen Sie die Aktion im [Visual Editor](#) aus und suchen Sie nach Feldern, in denen Informationen fehlen, oder schauen Sie in der Dokumentation zu Ihrer Aktion nach, auf die von verlinkt wird [Hinzufügen einer Aktion](#).
- Stellen Sie sicher, dass alle Eingabeartefakte über entsprechende Ausgabeartefakte verfügen. Weitere Informationen finden Sie unter [Definition eines Ausgabeartefakts](#).
- Stellen Sie sicher, dass die in einer Aktion definierten Variablen exportiert werden, damit sie in anderen Aktionen verwendet werden können. Weitere Informationen finden Sie unter [Eine Variable exportieren, sodass sie von anderen Aktionen verwendet werden kann](#).

## Wie behebe ich die Fehler „Anmeldeinformationen konnten nicht gefunden werden“ und `ExpiredToken` „?“

Problem: Während der [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#) Bearbeitung werden im Terminalfenster Ihres Entwicklungscomputers eine oder beide der folgenden Fehlermeldungen angezeigt:

```
Unable to locate credentials. You can configure credentials by running "aws configure".
```


```
ExpiredToken: The security token included in the request is expired
```

Mögliche Lösungen:

Diese Fehler weisen darauf hin, dass die Anmeldeinformationen, die Sie für den Zugriff auf AWS Dienste verwenden, abgelaufen sind. Führen Sie in diesem Fall den `aws configure` Befehl nicht aus. Verwenden Sie stattdessen die folgenden Anweisungen, um Ihren AWS Zugriffsschlüssel und Ihr Sitzungstoken zu aktualisieren.

Um Ihren AWS Zugriffsschlüssel und Ihr Sitzungstoken zu aktualisieren

1. Stellen Sie sicher, dass Sie die URL, den Benutzernamen und das Passwort des AWS Zugriffsportals für den Benutzer haben, den Sie verwenden, um das Amazon EKS-Tutorial abzuschließen (`codecatalyst-eks-user`). Sie sollten diese Elemente konfiguriert haben, als Sie das Tutorial abgeschlossen [Schritt 1: Richten Sie Ihren Entwicklungscomputer ein](#) haben.

 Note

Wenn Ihnen diese Informationen nicht vorliegen, rufen Sie die `codecatalyst-eks-user` Detailseite im IAM Identity Center auf und wählen Sie `Passwort zurücksetzen`, `Einmalpasswort generieren [...]`, und klicken Sie erneut auf `Passwort zurücksetzen`, um die Informationen auf dem Bildschirm anzuzeigen.

2. Führen Sie eine der folgenden Aktionen aus:
  - Fügen Sie die URL des AWS Zugangsportals in die Adressleiste Ihres Browsers ein.

Oder

  - Aktualisieren Sie die AWS Access-Portalseite, falls sie bereits geladen ist.
3. Melden Sie sich mit dem Benutzernamen und dem Passwort `codecatalyst-eks-user` von an, falls Sie noch nicht angemeldet sind.
4. Wählen Sie AWS-Kontound wählen Sie dann den Namen des Benutzers, AWS-Konto dem Sie den `codecatalyst-eks-user` Benutzer und den Berechtigungssatz zugewiesen haben.
5. Wählen Sie neben dem Namen des Berechtigungssatzes (`codecatalyst-eks-permission-set`) die Option `Befehlszeilenzugriff` oder `programmatischer Zugriff` aus.
6. Kopieren Sie die Befehle in der Mitte der Seite. Sie sehen etwa wie folgt aus:

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
export AWS_SECRET_ACCESS_KEY="wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... wobei *Session-Token* eine lange zufällige Zeichenfolge ist.

7. Fügen Sie die Befehle in Ihre Terminal-Eingabeaufforderung auf Ihrem Entwicklungscomputer ein und drücken Sie die Eingabetaste.

Die neuen Schlüssel und das Sitzungstoken werden geladen.

Sie haben jetzt Ihre Anmeldeinformationen aktualisiert. Die `kubectl` Befehle `AWS CLI` `eksctl`, und sollten jetzt funktionieren.

## Wie behebe ich die Fehler „Es konnte keine Verbindung zum Server hergestellt werden“?

Problem: Während Sie das unter beschriebene Tutorial durcharbeiten [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#), wird im Terminalfenster Ihres Entwicklungscomputers eine Fehlermeldung ähnlich der folgenden angezeigt:

```
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
```

Mögliche Lösungen:

Dieser Fehler weist normalerweise darauf hin, dass die Anmeldeinformationen, die das `kubectl` Hilfsprogramm für die Verbindung mit Ihrem Amazon EKS-Cluster verwendet, abgelaufen sind. Um das Problem zu lösen, aktualisieren Sie die Anmeldeinformationen, indem Sie an der Terminal-Eingabeaufforderung den folgenden Befehl eingeben:

```
aws eks update-kubeconfig --name codecatalyst-eks-cluster --region us-west-2
```

Wobei gilt:

- *codecatalyst-eks-cluster* wird durch den Namen Ihres Amazon EKS-Clusters ersetzt.
- *us-west-2* wird durch die AWS Region ersetzt, in der Ihr Cluster bereitgestellt wird.

## Warum fehlen CodeDeploy Felder im Visual Editor?

Problem: Sie verwenden eine Aktion „In [Amazon ECS bereitstellen](#)“ und die CodeDeploy Felder werden nicht angezeigt, z. B. CodeDeploy AppSpec im visuellen Editor des Workflows. Dieses Problem kann auftreten, weil der Amazon ECS-Service, den Sie im Feld Service angegeben haben, nicht für Blau/Grün-Bereitstellungen konfiguriert ist.

## Mögliche Lösungen:

- Wählen Sie auf der Registerkarte Konfiguration der Aktion In Amazon ECS bereitstellen einen anderen Amazon ECS-Service aus. Weitere Informationen finden Sie unter [Aktion „In Amazon ECS bereitstellen“ hinzufügen](#).
- Konfigurieren Sie den ausgewählten Amazon ECS-Service für Blau/Grün-Bereitstellungen. Weitere Informationen zur Konfiguration von Blue/Green-Bereitstellungen finden Sie unter [Blue/Green-Bereitstellung mit CodeDeploy](#) im Amazon Elastic Container Service Developer Guide.

## Wie behebe ich Fehler bei den IAM-Fähigkeiten?

Problem: Sie verwenden eine Aktion „[AWS CloudFormation Stack bereitstellen](#)“ und dies wird `##[error] requires capabilities: [capability-name]` in den Protokollen Ihrer Aktion „AWS CloudFormation Stack bereitstellen“ angezeigt.

Mögliche Lösungen: Gehen Sie wie folgt vor, um die Funktion zur Workflow-Definitionsdatei hinzuzufügen. Weitere Informationen zu IAM-Funktionen finden Sie unter [Bestätigung von IAM-Ressourcen in AWS CloudFormation Vorlagen im IAM-Benutzerhandbuch](#).

### Visual

So fügen Sie eine IAM-Funktion mit dem Visual Editor hinzu

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie Visual.
7. Wählen Sie im Workflow-Diagramm die Aktion `AWS CloudFormation Stack bereitstellen` aus.
8. Wählen Sie die Registerkarte Konfiguration aus.
9. Wählen Sie unten Erweitert — optional aus.



10. Aktivieren Sie in der Dropdownliste Funktionen das Kontrollkästchen neben der in der Fehlermeldung genannten Funktion. Wenn die Funktion in der Liste nicht verfügbar ist, fügen Sie sie mit dem YAML-Editor hinzu.
11. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
12. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.
13. Wenn eine neue Workflow-Ausführung nicht automatisch gestartet wird, führen Sie den Workflow manuell aus, um zu überprüfen, ob der Fehler durch die Änderungen behoben wird. Weitere Informationen zum manuellen Ausführen eines Workflows finden Sie unter [Einen Workflow-Lauf starten](#).

## YAML

So fügen Sie eine IAM-Fähigkeit mit dem YAML-Editor hinzu

1. [Öffnen Sie die CodeCatalyst Konsole unter https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Wählen Sie Ihr Projekt.
3. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
4. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie YAML.
7. Fügen Sie in der Aktion `AWS CloudFormation Stack bereitstellen` eine `capabilities` Eigenschaft wie die folgende hinzu:

```
DeployCloudFormationStack:  
  Configuration:  
    capabilities: capability-name
```

Ersetzen Sie *Capability-Name* durch den Namen der IAM-Fähigkeit, der in der Fehlermeldung angezeigt wird. Verwenden Sie Kommas und keine Leerzeichen, um mehrere Funktionen aufzulisten. Weitere Informationen finden Sie in der Beschreibung der `capabilities` Immobilie im [Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“](#).

8. (Optional) Wählen Sie „Validieren“, um den YAML-Code des Workflows vor dem Commit zu überprüfen.
9. Wählen Sie Commit, geben Sie eine Commit-Nachricht ein und wählen Sie erneut Commit aus.
10. Wenn eine neue Workflow-Ausführung nicht automatisch gestartet wird, führen Sie den Workflow manuell aus, um zu überprüfen, ob der Fehler durch die Änderungen behoben wird. Weitere Informationen zum manuellen Ausführen eines Workflows finden Sie unter [Einen Workflow-Lauf starten](#).

## Wie behebe ich „npm install“ -Fehler?

Problem: Ihre [AWS CDK Bereitstellungs- oder AWS CDK Bootstrap-Aktion](#) schlägt mit einem Fehler fehl. `npm install` Dieser Fehler kann auftreten, weil Sie Ihre AWS CDK App-Abhängigkeiten in einer NPM-Registrierung (Private Node Package Manager) speichern, auf die die Aktion nicht zugreifen kann.

Mögliche Lösungen: Verwenden Sie die folgenden Anweisungen, um die `cdk.json` Datei Ihrer AWS CDK App mit zusätzlichen Registrierungs- und Authentifizierungsinformationen zu aktualisieren.

Bevor Sie beginnen

1. Erstellen Sie Geheimnisse für Ihre Authentifizierungsinformationen. Sie werden in der `cdk.json` Datei auf diese Geheimnisse verweisen, anstatt die Entsprechungen im Klartext anzugeben. Um die Geheimnisse zu erstellen:
  - a. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
  - b. Wählen Sie Ihr Projekt.
  - c. Wählen Sie im Navigationsbereich CI/CD und dann Secrets aus.
  - d. Erstellen Sie zwei Geheimnisse mit den folgenden Eigenschaften:

| Erstes Geheimnis   | Zweites Geheimnis   |
|--|---|
| Name (Name: <code>npmUsername</code> )   | Name (Name: <code>npmAuthToken</code> )   |
| Wert: <code>npm-username</code> , wobei <code>npm-username</code> der <i>Benutzername</i> ist, | Wert: <code>npm-auth-token</code> , wobei sich das Zugriffstoken <code>npm-auth-token</code> befindet, das zur Authentifizierung bei Ihrer privaten |

| Erstes Geheimnis   | Zweites Geheimnis  |
|--|--|
| <p>der zur Authentifizierung bei Ihrer privaten npm-Registrierung verwendet wird.</p> <p>(Optional) Beschreibung: The username used to authenticate to the private npm registry.</p> | <p>NPM-Registrierung verwendet wird. Weitere Informationen zu NPM-Zugriffstoken finden Sie unter <a href="#">Über Zugriffstoken</a> in der npm-Dokumentation.</p> <p>(Optional) Beschreibung: The access token used to authenticate to the private npm registry.</p> |

Weitere Hinweise zu Geheimnissen finden Sie unter [Arbeiten mit Secrets](#).

2. Fügen Sie die Geheimnisse als Umgebungsvariablen zu Ihrer AWS CDK Aktion hinzu. Die Aktion ersetzt die Variablen durch echte Werte, wenn sie ausgeführt wird. Um die Geheimnisse hinzuzufügen:
  - a. Wählen Sie im Navigationsbereich CI/CD und dann Workflows aus.
  - b. Wählen Sie den Namen Ihres Workflows. Sie können nach dem Quell-Repository oder dem Branch-Namen filtern, in dem der Workflow definiert ist, oder nach dem Workflow-Namen filtern.
  - c. Wählen Sie Bearbeiten aus.
  - d. Wählen Sie Visual.
  - e. Wählen Sie im Workflow-Diagramm Ihre AWS CDK Aktion aus.
  - f. Wählen Sie die Registerkarte Eingaben.
  - g. Fügen Sie zwei Variablen mit den folgenden Eigenschaften hinzu:

| Erste Variable                                     | Zweite Variable                                     |
|--|---|
| Name (Name: NPMUSER)                               | Name (Name: NPMTOKEN)                               |
| Value (Wert): <code>\${Secrets.npmUsername}</code> | Value (Wert): <code>\${Secrets.npmAuthToken}</code> |

Sie haben jetzt zwei Variablen, die Verweise auf Geheimnisse enthalten.

Der YAML-Code Ihrer Workflow-Definitionsdatei sollte etwa wie folgt aussehen:

 Note

Das folgende Codebeispiel stammt aus einer AWS CDK Bootstrap-Aktion; eine AWS CDK Bereitstellungsaktion sieht ähnlich aus.


```
Name: CDK_Bootstrap_Action
SchemaVersion: 1.0
Actions:
  CDKBootstrapAction:
    Identifier: aws/cdk-bootstrap@v1
    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"
```

Sie sind jetzt bereit, die NPMTOKEN Variablen NPMUSER und in Ihrer `cdk.json` Datei zu verwenden. Fahren Sie mit dem nächsten Schritt fort.

Um deine `cdk.json`-Datei zu aktualisieren

1. Wechseln Sie in das Stammverzeichnis Ihres AWS CDK Projekts und öffnen Sie die `cdk.json` Datei.


- Suchen Sie die "app" : Eigenschaft und ändern Sie sie so, dass sie den *rot kursiv gedruckten* Code enthält:

 Note

Der folgende Beispielcode stammt aus einem TypeScript Projekt. Wenn Sie ein JavaScript Projekt verwenden, sieht der Code ähnlich, wenn auch nicht identisch aus.

```
{
  "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",
    ]
  },
  ...
}
```

- Ersetzen Sie in dem *rot kursiv* markierten Code:
  - Your-registry/folder/cdk-package/* mit dem Pfad zu Ihren Projektabhängigkeiten in Ihrer privaten Registrierung. AWS CDK
  - hello-cdk.ts|js mit* dem Namen Ihrer Einstiegsdatei. Dies kann je nach verwendeter Sprache eine .ts (TypeScript) oder .js (JavaScript) -Datei sein.

 Note

Die Aktion ersetzt die Variablen **NPMUSER** und **NPMTOKEN** durch den npm-Benutzernamen und das Zugriffstoken, die Sie in Secrets angegeben haben.

- Speichern Sie Ihre `cdk.json`-Datei.
- Führen Sie die Aktion manuell erneut aus, um festzustellen, ob der Fehler durch die Änderungen behoben wird. Weitere Informationen zum manuellen Ausführen von Aktionen finden Sie unter [Einen Workflow-Lauf starten](#).

## Warum haben mehrere Workflows denselben Namen?

Workflows werden pro Zweig pro Repository gespeichert. Zwei verschiedene Workflows können denselben Namen haben, wenn sie in verschiedenen Branches existieren. Auf der Workflow-Seite können Sie Workflows mit demselben Namen anhand des Branchennamens unterscheiden. Weitere Informationen finden Sie unter [Zusammenarbeit mit Filialen bei Amazon CodeCatalyst](#).

The screenshot shows the 'Workflows (6)' section of the Amazon CodeCatalyst interface. At the top right is a 'Create workflow' button. Below it is a search bar labeled 'Filter workflows'. Two workflow cards are displayed:

- The first card is for a workflow named 'BuildToProd' associated with the repository 'cfn-source-repository' and the branch 'branch-A'. It shows 'No recent runs' and a 'View all runs' link.
- The second card is also for a workflow named 'BuildToProd' associated with the repository 'cfn-source-repository' and the branch 'main'. It shows 'Recent runs' with four status icons (one green, one red, one green, one green) and a 'View all runs' link.

## Kann ich meine Workflow-Definitionsdateien in einem anderen Ordner speichern?

Nein, Sie müssen alle Workflow-Definitionsdateien in dem `.codecatalyst/workflows` Ordner speichern. Wenn Sie ein Mono-Repo mit mehreren logischen Projekten verwenden, platzieren Sie alle Ihre Workflow-Definitionsdateien in dem `.codecatalyst/workflows` Ordner und verwenden Sie dann die `FilesChanged` Eigenschaft in einem Trigger, um den Workflow an einem angegebenen Projektpfad auszulösen. Weitere Informationen finden Sie unter [Arbeiten mit Auslösern](#).

## Wie füge ich meinem Workflow nacheinander Aktionen hinzu?

Wenn Sie Ihrem Workflow eine Aktion hinzufügen, hat sie standardmäßig keine Abhängigkeiten und wird parallel zu anderen Aktionen ausgeführt.

Wenn Sie Aktionen der Reihe nach anordnen möchten, können Sie eine Abhängigkeit von einer anderen Aktion festlegen, indem Sie das `DependsOn` Feld festlegen. Sie können eine Aktion auch so konfigurieren, dass sie Artefakte oder Variablen verwendet, die Ergebnisse anderer Aktionen sind. Weitere Informationen finden Sie unter [Aktionen so konfigurieren, dass sie von anderen Aktionen abhängen](#).

## Warum wird mein Workflow erfolgreich validiert, schlägt aber zur Laufzeit fehl?

Wenn Sie Ihren Workflow mit der `Validate` Schaltfläche validiert haben, Ihr Workflow aber trotzdem fehlgeschlagen ist, kann das an einer Einschränkung des Validators liegen.

Fehler in Bezug auf eine CodeCatalyst Ressource wie Geheimnisse, Umgebungen oder Flotten in der Workflow-Konfiguration werden bei einem Commit nicht registriert. Wenn ungültige Verweise verwendet werden, wird der Fehler nur erkannt, wenn ein Workflow ausgeführt wird. Ebenso werden Fehler in Ihrer Aktionskonfiguration, wie z. B. das Fehlen eines erforderlichen Felds oder Tippfehler in den Aktionsattributen, erst erkannt, wenn der Workflow ausgeführt wird. Weitere Informationen finden Sie unter [Einen Workflow erstellen, bearbeiten und löschen](#).

## Die automatische Erkennung erkennt keine Berichte für meine Aktion

**Problem:** Ich habe Autodiscovery für eine Aktion konfiguriert, die Tests ausführt, aber es werden keine Berichte von CodeCatalyst erkannt.

**Mögliche Lösungen:** Dies kann durch eine Reihe von Problemen verursacht werden. Probieren Sie eine oder mehrere der folgenden Lösungen aus:

- Stellen Sie sicher, dass das zum Ausführen von Tests verwendete Tool Ausgaben in einem der Formate erzeugt, die ich CodeCatalyst verstehe. Wenn Sie beispielsweise zulassen CodeCatalyst möchten, dass Berichte `pytest` zur Test- und Codeabdeckung erkannt werden, geben Sie die folgenden Argumente an:

```
--junitxml=test_results.xml --cov-report xml:test_coverage.xml
```

Weitere Informationen finden Sie unter [Typen von Testberichten](#).

- Stellen Sie sicher, dass die Dateierweiterungen für die Ausgaben dem ausgewählten Format entsprechen. Wenn Sie beispielsweise so konfigurieren `pytest`, dass Ergebnisse im JUnitXML Format angezeigt werden, überprüfen Sie, ob die `.xml` Dateierweiterung Weitere Informationen finden Sie unter [Typen von Testberichten](#).
- Stellen Sie sicher, dass `IncludePaths` es so konfiguriert ist, dass es das gesamte Dateisystem (`**/*`) einschließt, es sei denn, Sie schließen absichtlich bestimmte Ordner aus. Achten Sie auch darauf, dass `ExcludePaths` Sie keine Verzeichnisse ausschließen, in denen sich Ihre Berichte voraussichtlich befinden werden.
- Wenn Sie einen Bericht manuell für die Verwendung einer bestimmten Ausgabe datei konfiguriert haben, wird er von der automatischen Erkennung ausgeschlossen. Weitere Informationen finden Sie unter [Beispiel: Berichte konfigurieren](#).
- Bei der automatischen Erkennung werden möglicherweise keine Berichte gefunden, da die Aktion fehlgeschlagen ist, bevor Ausgaben generiert wurden. Beispielsweise ist der Build möglicherweise fehlgeschlagen, bevor Komponententests ausgeführt wurden.

## Meine Aktion schlägt bei automatisch erkannten Berichten fehl, nachdem ich Erfolgskriterien konfiguriert habe

Problem: Wenn ich die automatische Erkennung aktiviere und Erfolgskriterien konfiguriere, erfüllen einige Berichte die Erfolgskriterien nicht und die Aktion schlägt daher fehl.

Mögliche Lösungen: Probieren Sie eine oder mehrere der folgenden Lösungen aus, um dieses Problem zu beheben:

- Ändern Sie `BerichteExcludePaths`, an denen Sie nicht interessiert sind, `IncludePaths` oder schließen Sie sie aus.
- Aktualisieren Sie die Erfolgskriterien, damit alle Berichte erfolgreich sind. Wenn beispielsweise zwei Berichte gefunden wurden, von denen einer eine Leitungsabdeckung von 50% und der andere von 70% aufweist, passen Sie die Mindestleitungsabdeckung auf 50% an. Weitere Informationen finden Sie unter [Erfolgskriterien](#).
- Verwandeln Sie den fehlgeschlagenen Bericht in einen manuell konfigurierten Bericht. Auf diese Weise können Sie verschiedene Erfolgskriterien für diesen bestimmten Bericht konfigurieren. Weitere Informationen finden Sie unter [Erfolgskriterien für Berichte konfigurieren](#).



## Autodiscovery generiert Berichte, die ich nicht haben möchte

**Problem:** Wenn ich die automatische Erkennung aktiviere, werden Berichte generiert, die ich nicht möchte. CodeCatalyst Generiert beispielsweise Berichte zur Codeabdeckung für Dateien, die in den Abhängigkeiten meiner Anwendung enthalten sind, die in `node_modules` gespeichert sind.

**Mögliche Lösungen:** Sie können die `ExcludePaths` Konfiguration so anpassen, dass unerwünschte Dateien ausgeschlossen werden. Um beispielsweise auszuschließen `node_modules`, fügen Sie hinzu `node_modules/**/*`. Weitere Informationen finden Sie unter [Pfade einschließen/ausschließen](#).

## Auto-Discovery generiert viele kleine Berichte für ein einzelnes Test-Framework

**Problem:** Bei der Verwendung bestimmter Frameworks für Test- und Codeabdeckungsberichte habe ich festgestellt, dass Autodiscovery eine große Anzahl von Berichten generiert. Wenn Sie beispielsweise das [Maven Surefire-Plugin](#) verwenden, erstellt die automatische Erkennung für jede Testklasse einen anderen Bericht.

**Mögliche Korrekturen:** Ihr Framework kann möglicherweise Ausgaben in einer einzigen Datei zusammenfassen. Wenn Sie beispielsweise das Maven Surefire Plugin verwenden, können Sie es verwenden, um die Dateien `npx junit-merge` manuell zu aggregieren. Der vollständige Ausdruck könnte so aussehen:

```
mvn test; cd test-package-path/surefire-reports && npx junit-merge -d ./ && rm
*Test.xml
```

## Die unter CI/CD aufgeführten Workflows stimmen nicht mit denen im Quell-Repository überein

**Problem:** [Die auf der Seite CI/CD, Workflows angezeigten Workflows stimmen nicht mit denen im ~/.codecatalyst/workflows/ Ordner in Ihrem Quell-Repository überein](#). Möglicherweise sehen Sie die folgenden Diskrepanzen:

- Ein Workflow wird auf der Workflow-Seite angezeigt, aber eine entsprechende Workflow-Definitionsdatei ist in Ihrem Quell-Repository nicht vorhanden.
- In Ihrem Quell-Repository ist eine Workflow-Definitionsdatei vorhanden, aber ein entsprechender Workflow wird nicht auf der Workflow-Seite angezeigt.

- Ein Workflow ist sowohl im Quell-Repository als auch auf der Workflow-Seite vorhanden, aber die beiden sind unterschiedlich.

Dieses Problem kann auftreten, wenn die Workflow-Seite nicht aktualisiert werden konnte oder wenn ein Workflow-Kontingent überschritten wurde.

Mögliche Lösungen:

- Warten Sie. Normalerweise müssen Sie nach einem Commit an die Quelle zwei oder drei Sekunden warten, bis Sie die Änderung auf der Workflow-Seite sehen.
- Wenn Sie ein Workflow-Kontingent überschritten haben, gehen Sie wie folgt vor:

#### Note

Um festzustellen, ob ein Workflow-Kontingent überschritten wurde [Kontingente für Workflows in CodeCatalyst](#), überprüfen Sie die dokumentierten Kontingente und vergleichen Sie sie mit den Workflows in Ihrem Quell-Repository oder auf der Workflow-Seite. Es gibt keine Fehlermeldung, die darauf hinweist, dass ein Kontingent überschritten wurde. Sie müssen es also selbst untersuchen.

- Wenn Sie die maximale Anzahl von Workflows pro Speicherkontingent überschritten haben, [löschen Sie einige Workflows](#) und führen Sie dann einen Test-Commit anhand der Workflow-Definitionsdatei durch. Ein Beispiel für ein Test-Commit könnte darin bestehen, der Datei ein Leerzeichen hinzuzufügen.
- Wenn Sie das Kontingent für die maximale Größe der Workflow-Definitionsdatei überschritten haben, ändern Sie die Länge der Workflow-Definitionsdatei.
- Wenn Sie die maximale Anzahl von Workflow-Dateien, die in einer einzelnen Quelle verarbeitet werden, überschritten haben, führen Sie mehrere Test-Commits durch. Ändern Sie weniger als die maximale Anzahl von Workflows pro Commit.
- Erhöhen Sie die Workflow-Kontingente, indem Sie die Abrechnung für kostenpflichtige Tarife aktivieren. Weitere Informationen finden Sie unter [Abrechnung verwalten](#) im CodeCatalyst Amazon-Administratorhandbuch.

## Ich kann keine Workflows erstellen oder aktualisieren

**Problem:** Ich möchte einen Workflow erstellen oder aktualisieren, erhalte aber eine Fehlermeldung, wenn ich versuche, die Änderung zu übernehmen.

**Mögliche Lösungen:** Abhängig von Ihrer Rolle im Projekt oder Space verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen, um Code in Quell-Repositorys im Projekt zu übertragen. Die YAML-Dateien für Workflows werden in Repositorys gespeichert. Weitere Informationen finden Sie unter [Workflow-Definitionsdateien](#). Die Rollen „Space-Administrator“, „Projektadministrator“ und „Mitwirkender“ sind alle berechtigt, Code in Repositorys in einem Projekt zu übertragen und dort per Push zu speichern.

Wenn Sie die Rolle Mitwirkender haben, aber in einem bestimmten Branch keine Änderungen an Workflow-YAML erstellen oder übernehmen können, ist möglicherweise für diesen Branch eine Branch-Regel konfiguriert, die verhindert, dass Benutzer mit dieser Rolle Code in diesen bestimmten Branch pushen. Versuchen Sie, einen Workflow in einem anderen Zweig zu erstellen oder Ihre Änderungen an einen anderen Zweig zu übertragen. Weitere Informationen finden Sie unter [Zulässige Aktionen für einen Branch mit Branch-Regeln verwalten](#).

## Behebung von Problemen mit der Suche in CodeCatalyst

Lesen Sie die folgenden Abschnitte, um Probleme im Zusammenhang mit der Suche in zu beheben CodeCatalyst. Weitere Informationen zu Workflows finden Sie unter [Suche in CodeCatalyst](#).

### Themen

- [Ich kann in meinem Projekt keinen Benutzer finden](#)
- [Ich sehe in meinem Projekt oder Bereich nicht, wonach ich suche](#)
- [Die Anzahl der Suchergebnisse ändert sich ständig, wenn ich durch die Seiten navigiere](#)
- [Meine Suchabfrage wird nicht abgeschlossen](#)

## Ich kann in meinem Projekt keinen Benutzer finden

**Problem:** Wenn ich versuche, die Details eines Benutzers anzuzeigen, sehe ich seine Informationen nicht im Projekt.

**Mögliche Korrekturen:** Die Suche unterstützt derzeit nicht die Suche nach Benutzern innerhalb eines Projekts. Um nach Benutzern zu suchen, die Zugriff auf Ihren Bereich haben, wechseln Sie zu Dieser

Bereich in oder entfernen Sie alle Projektfilter QuickSearch, die Sie möglicherweise mithilfe der erweiterten Abfragesprache angegeben haben.

## Ich sehe in meinem Projekt oder Bereich nicht, wonach ich suche

**Problem:** Wenn ich versuche, nach bestimmten Informationen zu suchen, werden keine Ergebnisse angezeigt.

**Mögliche Lösungen:** Es dauert wahrscheinlich einige Sekunden, bis Inhaltsaktualisierungen in den Suchergebnissen aktualisiert werden. Umfangreiche Updates können mehrere Minuten dauern.

Bei Ressourcen, die in letzter Zeit nicht aktualisiert wurden, müssen Sie möglicherweise Ihre Suche verfeinern. Sie können Ihre Suche verfeinern, indem Sie weitere Stichwörter hinzufügen oder die erweiterte Abfragesprache verwenden. Weitere Informationen zur Verfeinerung Ihrer Abfragen finden Sie unter [Verfeinern Sie Ihre Suchabfrage](#).

## Die Anzahl der Suchergebnisse ändert sich ständig, wenn ich durch die Seiten navigiere

**Problem:** Die Anzahl der Suchergebnisse scheint sich zu ändern, wenn ich zur nächsten Seite gehe, sodass nicht klar ist, wie viele Ergebnisse es insgesamt gibt.

**Mögliche Lösungen:** Wenn Sie durch Seiten mit Suchergebnissen navigieren, sehen Sie möglicherweise eine Änderung der Anzahl der Suchergebnisse, die Ihrer Anfrage entsprechen. Die Anzahl der Ergebnisse wird möglicherweise aktualisiert, um eine genauere Anzahl der Treffer wiederzugeben, die beim Navigieren durch die Seiten gefunden wurden.

Wenn Sie durch die Ergebnisse navigieren, wird möglicherweise die folgende Meldung angezeigt: Keine Ergebnisse für „Test“. Sie erhalten die Nachricht, wenn Sie keinen Zugriff auf die verbleibenden Ergebnisse haben.

## Meine Suchabfrage wird nicht abgeschlossen

**Problem:** Die Ergebnisse meiner Suchabfrage werden nicht angezeigt und es scheint zu lange zu dauern.

**Mögliche Lösungen:** Ihre Suche wird möglicherweise nicht abgeschlossen, wenn in dem Bereich viele Suchanfragen gleichzeitig durchgeführt werden, entweder programmgesteuert oder aufgrund hoher

Teamaktivität. Wenn Sie programmatische Suchen ausführen, pausieren oder reduzieren Sie sie. Andernfalls versuchen Sie es in ein paar Sekunden erneut.

## Beheben von Problemen mit Konten, die Ihrem Bereich zugeordnet sind

In können Sie AWS-Konto Ihrem Bereich eine hinzufügen CodeCatalyst, um Ressourcen und Fakturierungszwecke Berechtigungen zu erteilen. Die folgenden Informationen können Ihnen bei der Behebung häufiger Probleme mit zugehörigen Konten in helfen CodeCatalyst.

### Themen

- [Meine AWS-Konto Verbindungsanforderung erhält einen ungültigen Token-Fehler](#)
- [Mein Amazon- CodeCatalyst Projektworkflow schlägt mit einem Fehler für das konfigurierte Konto, die Umgebung oder die IAM-Rolle fehl](#)
- [Ich benötige ein zugeordnetes Konto, eine Rolle und eine Umgebung, um ein Projekt zu erstellen](#)
- [Ich kann nicht auf die Seite Amazon CodeCatalyst Spaces in der zugreifen AWS Management Console](#)
- [Ich möchte ein anderes Konto als mein Abrechnungskonto](#)

## Meine AWS-Konto Verbindungsanforderung erhält einen ungültigen Token-Fehler

Problem: Beim Erstellen einer Verbindungsanforderung mit einem Verbindungstoken akzeptiert die Seite das Token nicht und zeigt einen Fehler an, der besagt, dass das Token ungültig ist.

Mögliche Korrekturen: Stellen Sie sicher, dass Sie die Konto-ID angeben, die Sie Ihrem Speicherplatz hinzufügen möchten. Sie müssen über Administratorberechtigungen für Ihr verfügen AWS-Konto oder mit Ihrem Administrator zusammenarbeiten können, um das Konto hinzuzufügen.

Wenn Sie das Konto verifizieren möchten, wird ein neues Browserfenster in der geöffnetAWS Management Console. Dasselbe Konto muss auf der Konsolenseite angemeldet sein. Versuchen Sie es erneut, nachdem Sie Folgendes überprüft haben:

- Sie sind bei der AWS Management Console mit derselben angemeldetAWS-Konto, die Sie Ihrem Bereich hinzufügen möchten.

- Sie sind bei der angemeldet, AWS Management Console wobei USA West (Oregon) AWS-Region (us-west-2) ausgewählt ist.
- Wenn Sie von der Fakturierungsseite angekommen sind und die AWS-Konto als angegebenes Fakturierungskonto für Ihren Speicherplatz hinzufügen möchten, stellen Sie sicher, dass das Konto noch kein Fakturierungskonto für einen anderen Speicherplatz ist.

## Mein Amazon- CodeCatalyst Projektworkflow schlägt mit einem Fehler für das konfigurierte Konto, die Umgebung oder die IAM-Rolle fehl

Problem: Wenn der Workflow ausgeführt wird und kein konfiguriertes Konto oder keine IAM-Rollen findet, die Ihrem Bereich zugeordnet sind, müssen Sie die Felder Rolle, Verbindung und Umgebung manuell im Workflow YAML ausfüllen. Sehen Sie sich die fehlgeschlagene Workflow-Aktion an und notieren Sie sich, ob die Fehlermeldungen wie folgt lauten:

- Die Rolle ist nicht für die Verwendung mit der Verbindung verfügbar, die der Umgebung zugeordnet ist.
- Die Aktion war nicht erfolgreich. Status: FAILED; Der angegebene Wert für die Kontoverbindung oder Umgebung ist ungültig. Überprüfen Sie, ob die Verbindung mit Ihrem Bereich und die Umgebung mit Ihrem Projekt verknüpft ist.
- Die Aktion war nicht erfolgreich. Status: FAILED; Der angegebene Wert für die IAM-Rolle ist ungültig. Überprüfen Sie, ob der Name vorhanden ist, die IAM-Rolle zu Ihrer Kontoverbindung hinzugefügt wurde und die Verbindung bereits mit Ihrem Amazon- CodeCatalyst Bereich verknüpft ist

Mögliche Korrekturen: Stellen Sie sicher, dass die Workflow-YAML-Felder genaue Werte für [Umgebung](#) , [Verbindungen](#) und [Rolle](#) haben. Die CodeCatalyst Workflow-Aktionen, die eine Umgebung erfordern, sind Build- oder Bereitstellungsaktionen, die -AWSRessourcen ausführen oder AWS Ressourcen-Stacks generieren.

Wählen Sie den Block für fehlgeschlagene Workflow-Aktionen und dann Visual aus. Wählen Sie die Registerkarte Konfiguration aus. Wenn die Felder Umgebung , Verbindungsname und Rollenname nicht ausgefüllt sind, müssen Sie den Workflow manuell aktualisieren. Gehen Sie wie folgt vor, um Ihren Workflow-YAML zu bearbeiten:

- Erweitern Sie das `./codecatalyst` Verzeichnis und dann das `./workflows` Verzeichnis. Öffnen Sie die Workflow-YAML-Datei. Stellen Sie sicher, dass die IAM-Rollen und Kontoinformationen in der YAML angegeben sind, die Sie für Ihren Workflow konfiguriert haben. Beispiel:

```
Actions:
  cdk_bootstrap:
    Identifier: action-@v1
    Inputs:
      Sources:
        - WorkflowSource
    Environment:
      Name: Staging
    Connections:
      - Name: account-connection
        Role: build-role
```

Die Eigenschaften Umgebung, Verbindung und Rolle sind erforderlich, um CodeCatalyst Workflow-Build- und Bereitstellungsaktionen mit -AWS-Ressourcen auszuführen. Ein Beispiel finden Sie in der YAML-Referenzparameter der CodeCatalyst Build-Aktion für [Umgebung](#), [Verbindungen](#) und [Rolle](#).

- Stellen Sie sicher, dass Ihrem Speicherplatz ein Konto hinzugefügt wurde, und stellen Sie sicher, dass dem Konto die entsprechende IAM-Rolle oder Rollen hinzugefügt wurden. Sie können Konten anpassen oder hinzufügen, wenn Sie über die Space-Administratorrolle verfügen. Weitere Informationen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

## Ich benötige ein zugeordnetes Konto, eine Rolle und eine Umgebung, um ein Projekt zu erstellen

**Problem:** In den Optionen zur Projekterstellung ist für mein Projekt entweder kein zusätzliches Konto in meinem Bereich verfügbar, oder ich benötige ein anderes Konto, das meinem Bereich hinzugefügt wird, damit mein Projekt es verwenden kann.

**Mögliche Korrekturen:** Für Ihren Bereich können Sie -autorisierte hinzufügen, AWS-Konten um sie Ihrem Projekt hinzuzufügen, wenn Sie über die Rolle Space-Administrator verfügen. Sie müssen auch über ein verfügbares AWS-Konto, in dem Sie über Administratorberechtigungen verfügen oder mit Ihrem AWS Administrator zusammenarbeiten können.

Um sicherzustellen, dass ein Konto und eine Rolle auf dem Projekterstellungsbildschirm verfügbar sind, müssen Sie zuerst das Konto und die Rollen hinzufügen. Weitere Informationen finden Sie unter [Verwaltung AWS-Konten für einen Raum](#).

Sie haben die Möglichkeit, eine Servicerolle mit einer Rollenrichtlinie zu erstellen, die als CodeCatalystWorkflowDevelopmentRole-*spaceName* Rollenrichtlinie bezeichnet wird. Der Rolle wird ein Name CodeCatalystWorkflowDevelopmentRole-*spaceName* mit einer eindeutigen Kennung angehängt. Weitere Informationen zur Rolle und zur Rollenrichtlinie finden Sie unter [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#). Die Schritte zum Erstellen der Rolle finden Sie unter [Erstellen der CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle für Ihr Konto und Ihren Bereich](#). Die Rolle wird Ihrem Konto hinzugefügt und ist auf den Seiten zur Projekterstellung in verfügbar CodeCatalyst.

## Ich kann nicht auf die Seite Amazon CodeCatalyst Spaces in der zugreifen AWS Management Console

Problem: Wenn ich versuche, auf die Amazon- CodeCatalyst Seite in der zuzugreifenAWS Management Console, um meinem CodeCatalyst Bereich ein Konto hinzuzufügen oder einem Konto in Rollen hinzuzufügenAWS, erhalte ich einen Berechtigungsfehler.

Mögliche Lösungen:

Für Ihren Bereich können Sie autorisiert hinzufügen, AWS-Konten um sie Ihrem Projekt hinzuzufügen, wenn Sie über die Space-Administratorrolle verfügen. Sie müssen auch über ein verfügenAWS-Konto, in dem Sie über Administratorberechtigungen verfügen oder mit Ihrem AWS Administrator zusammenarbeiten können. Sie müssen zunächst sicherstellen, dass Sie bei der AWS Management Console mit demselben Konto angemeldet sind, das Sie verwalten möchten. Nachdem Sie bei der angemeldet sindAWS Management Console, können Sie die -Konsole öffnen und es erneut versuchen.

Öffnen Sie die Amazon- CodeCatalyst Seite im AWS Management Console unter [https://us-west-2.console.aws.amazon.com/codecatalyst/home?region=us-west-2#](https://us-west-2.console.aws.amazon.com/codecatalyst/home?region=us-west-2#/).

## Ich möchte ein anderes Konto als mein Abrechnungskonto

Problem: Wenn ich meine CodeCatalyst Anmeldung eingerichtet habe, habe ich mehrere Schritte ausgeführt, um meinen Bereich einzurichten und eine autorisierte zuzuordnenAWS-Konto. Jetzt möchte ich ein anderes Konto für die Abrechnung autorisieren.



Mögliche Korrekturen: Für Ihren Speicherplatz können Sie Fakturierungskonten autorisieren, wenn Sie über die Rolle Space-Administrator verfügen. Sie müssen auch über ein verfügbares AWS-Konto, in dem Sie über Administratorberechtigungen verfügen oder mit Ihrem AWS Administrator zusammenarbeiten können.

Weitere Informationen finden Sie unter [Verwalten der Fakturierung](#) im Amazon- CodeCatalyst Administratorhandbuch.

## Behebung von Problemen mit Entwicklungsumgebungen

Lesen Sie die folgenden Abschnitte, um Probleme im Zusammenhang mit Entwicklungsumgebungen zu beheben. Weitere Informationen zu Entwicklungsumgebungen finden Sie unter [Entwicklungsumgebungen in CodeCatalyst](#).

### Themen

- [Die Erstellung meiner Entwicklungsumgebung war aufgrund eines Problems mit den Kontingenten nicht erfolgreich](#)
- [Ich kann keine Änderungen aus meiner Entwicklungsumgebung in einen bestimmten Zweig in einem Repository übertragen](#)
- [Meine Entwicklungsumgebung wurde nicht wieder aufgenommen](#)
- [Die Verbindung zu meiner Entwicklungsumgebung wurde unterbrochen](#)
- [Meine mit dem VPC verbundene Entwicklungsumgebung ist ausgefallen](#)
- [Ich kann nicht finden, in welchem Verzeichnis sich mein Projekt befindet](#)
- [Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen](#)
- [Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen, da meine lokale SSH-Konfiguration fehlt](#)
- [Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen, da ich Probleme mit meinem Profil AWS Config habe codecatalyst](#)
- [Fehlerbehebung bei Problemen mit IDEs](#)
- [Behebung von Problemen mit Devfiles](#)

## Die Erstellung meiner Entwicklungsumgebung war aufgrund eines Problems mit den Kontingenten nicht erfolgreich

**Problem:** Ich möchte eine Entwicklungsumgebung in erstellen CodeCatalyst, aber mir wird ein Fehler angezeigt. In der Konsole wird auf der Seite „Entwicklungsumgebungen“ eine Meldung angezeigt, dass ich das Speicherlimit für den Speicherplatz erreicht habe.

**Mögliche Lösungen:** Abhängig von Ihrer Rolle im Projekt oder Space können Sie entweder eine oder mehrere Ihrer eigenen Entwicklungsumgebungen löschen, oder wenn Sie die Rolle eines Space-Administrators haben, können Sie ungenutzte Entwicklungsumgebungen löschen, die von anderen Benutzern erstellt wurden. Sie können sich auch dafür entscheiden, die Abrechnungsstufe auf eine Stufe zu ändern, die mehr Speicherplatz beinhaltet.

- Um die Speicherlimits einzusehen, schauen Sie im CodeCatalyst Amazon-Bereich auf der Registerkarte Abrechnung nach, ob die Nutzungskontingente das zulässige Maximum erreicht haben. Wenn die Kontingente das Maximum erreicht haben, wenden Sie sich an jemanden mit der Rolle des Space-Administrators, um nicht benötigte Entwicklungsumgebungen zu entfernen oder eine Änderung der Abrechnungsstufe in Betracht zu ziehen.
- Informationen zum Entfernen von Entwicklungsumgebungen, die Sie nicht mehr benötigen, finden Sie unter [Löschen einer Entwicklungsumgebung](#).

Wenn das Problem weiterhin besteht und Sie einen Fehler in Ihrer IDE erhalten, überprüfen Sie, ob Sie über eine CodeCatalyst Rolle verfügen, mit der Sie eine Entwicklungsumgebung erstellen können. Die Rollen Space-Administrator, Projektadministrator und Mitwirkender verfügen alle über die Berechtigung, Entwicklungsumgebungen zu erstellen. Weitere Informationen finden Sie unter [Mit Rollen in Amazon arbeiten CodeCatalyst](#).

## Ich kann keine Änderungen aus meiner Entwicklungsumgebung in einen bestimmten Zweig in einem Repository übertragen

**Problem:** Ich möchte Codeänderungen in meiner Entwicklungsumgebung in einen Branch in einem Quell-Repository übertragen und per Push übertragen, aber es wird ein Fehler angezeigt.

**Mögliche Lösungen:** Abhängig von Ihrer Rolle im Projekt oder Bereich verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen, um Code in Quell-Repositorys im Projekt zu übertragen. Die Rollen Space-Administrator, Projektadministrator und Mitwirkender sind alle berechtigt, Code in Repositorys im Projekt zu pushen.

Wenn Sie die Rolle „Mitwirkender“ haben, aber keinen Code in einen bestimmten Zweig übertragen können, ist möglicherweise eine Verzweigungsregel für diesen bestimmten Zweig konfiguriert, die verhindert, dass Benutzer mit dieser Rolle Code in diesen bestimmten Zweig übertragen. Versuchen Sie, Ihre Änderungen in einen anderen Zweig zu übertragen, oder erstellen Sie einen Zweig und übertragen Sie dann Ihren Code in diesen Zweig. Weitere Informationen finden Sie unter [Zulässige Aktionen für einen Branch mit Branch-Regeln verwalten](#).

## Meine Entwicklungsumgebung wurde nicht wieder aufgenommen

**Problem:** Meine Entwicklungsumgebung wurde nicht wieder aufgenommen, nachdem ich sie beendet hatte.

**Mögliche Lösungen:** Um das Problem zu beheben, schauen Sie im CodeCatalyst Amazon-Bereich auf der Registerkarte Abrechnung nach, ob die Nutzungskontingente die Höchstgrenzen erreicht haben. Wenn die Kontingente das Höchstlimit erreicht haben, wenden Sie sich an Ihren Space-Administrator, um die Abrechnungsstufe zu erhöhen.

## Die Verbindung zu meiner Entwicklungsumgebung wurde unterbrochen

**Problem:** Die Verbindung zu meiner Entwicklungsumgebung wurde unterbrochen, während ich sie verwendet habe.

**Mögliche Lösungen:** Überprüfen Sie Ihre Internetverbindung, um das Problem zu beheben. Wenn Sie nicht mit dem Internet verbunden sind, stellen Sie eine Verbindung her und setzen Sie Ihre Arbeit in Ihrer Entwicklungsumgebung fort.

## Meine mit dem VPC verbundene Entwicklungsumgebung ist ausgefallen

**Problem:** Ich habe eine VPC-Verbindung mit meiner Entwicklungsumgebung verknüpft und es treten Fehler auf.

**Mögliche Lösungen:** Docker verwendet ein Link-Layer-Gerät, ein sogenanntes Bridge-Netzwerk, das es Containern ermöglicht, zu kommunizieren, die mit demselben Bridge-Netzwerk verbunden sind. Die Standard-Bridge verwendet normalerweise das 172.17.0.0/16 Subnetz für Container-Netzwerke. Wenn das VPC-Subnetz für die Instance Ihrer Umgebung denselben Adressbereich verwendet, der bereits von Docker genutzt wird, kann ein IP-Adressenkonflikt auftreten. Um einen IP-Adressenkonflikt zu lösen, der dadurch verursacht wird, dass Amazon VPC Docker denselben IPv4-CIDR-Adressblock verwendet, konfigurieren Sie einen anderen CIDR-Block als 172.17.0.0/16

**Note**

Sie können den IP-Adressbereich einer vorhandenen VPC oder eines Subnetzes nicht ändern.

## Ich kann nicht finden, in welchem Verzeichnis sich mein Projekt befindet

Problem: Ich kann nicht finden, in welchem Verzeichnis sich mein Projekt befindet.

Mögliche Lösungen: Um Ihr Projekt zu finden, wechseln Sie in das Verzeichnis/`projects`. Dies ist das Verzeichnis, in dem Sie Ihre Projekte finden können.

## Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen

Um Fehler bei der Verbindung zu Ihrer Entwicklungsumgebung über SSH zu beheben, können Sie den `ssh` Befehl mit der `-vvv` Option ausführen, um weitere Informationen zur Lösung Ihres Problems anzuzeigen:

```
ssh -vvv codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

## Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen, da meine lokale SSH-Konfiguration fehlt

Wenn Ihre lokale SSH-Konfiguration (`~/.ssh/config`) fehlt oder der Inhalt des Host `codecatalyst-dev-env*` Abschnitts veraltet ist, können Sie keine Verbindung zu Ihrer Entwicklungsumgebung über SSH herstellen. Um dieses Problem zu beheben, löschen Sie den Host `codecatalyst-dev-env*` Abschnitt und führen Sie den ersten Befehl aus dem SSH Access-Modal erneut aus. Weitere Informationen finden Sie unter [Verbindung zu einer Entwicklungsumgebung über SSH herstellen](#).

## Ich kann keine Verbindung zu meiner Entwicklungsumgebung über SSH herstellen, da ich Probleme mit meinem Profil AWS Config habe

### codecatalyst

Vergewissern Sie sich, dass Ihr AWS Config (~/.aws/config) für das codecatalyst Profil mit dem unter beschriebenen übereinstimmt [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#). Wenn nicht, löschen Sie das Profil für codecatalyst und führen Sie den ersten Befehl aus dem SSH Access-Modal erneut aus. Weitere Informationen finden Sie unter [Verbindung zu einer Entwicklungsumgebung über SSH herstellen](#).

## Fehlerbehebung bei Problemen mit IDEs

Lesen Sie die folgenden Abschnitte, um Probleme im Zusammenhang mit IDEs in zu beheben CodeCatalyst. Weitere Informationen zu IDEs finden Sie unter [Verwenden von Entwicklungsumgebungen mit einer IDE](#).

### Themen

- [Ich habe nicht übereinstimmende Laufzeit-Image-Versionen in AWS Cloud9](#)
- [Ich kann nicht auf meine Dateien in /projects/projects in zugreifen AWS Cloud9](#)
- [Ich kann meine Entwicklungsumgebung in nicht AWS Cloud9 mit einer benutzerdefinierten Devfile-Datei starten](#)
- [Ich habe Probleme in AWS Cloud9](#)
- [In kann JetBrainsich keine Verbindung zu meinen Entwicklungsumgebungen über herstellen CodeCatalyst](#)
- [Ich kann nicht AWS Toolkit für meine IDE installieren](#)
- [In meiner IDE kann ich meine Entwicklungsumgebungen nicht starten](#)

## Ich habe nicht übereinstimmende Laufzeit-Image-Versionen in AWS Cloud9

AWS Cloud9 verwendet verschiedene Versionen der Frontend-Komponente und des Backend-Laufzeit-Images. Die Verwendung verschiedener Versionen kann dazu führenAWS Toolkit, dass die Git-Erweiterung und falsch funktionieren. Um das Problem zu beheben, navigieren Sie zum Dashboard der Entwicklungsumgebung, halten Sie Ihre Entwicklungsumgebung an und starten Sie sie dann erneut. Um das Problem mithilfe von APIs zu beheben, verwenden Sie die

UpdateDevEnvironment-API, um die Laufzeit zu aktualisieren. Weitere Informationen finden Sie unter [UpdateDevEnvironment](#) in der Amazon CodeCatalyst -API-Referenz .

Ich kann nicht auf meine Dateien in `/projects/projects` in zugreifen AWS Cloud9

Der AWS Cloud9 Editor kann nicht auf Dateien im Verzeichnis zugreifen/`projects/projects`. Um das Problem zu beheben, verwenden Sie das AWS Cloud9 Terminal, um auf Ihre Dateien zuzugreifen oder sie in ein anderes Verzeichnis zu verschieben.

Ich kann meine Entwicklungsumgebung in nicht AWS Cloud9 mit einer benutzerdefinierten Devfile-Datei starten

Ihr Devfile-Image ist möglicherweise nicht mit kompatibelAWS Cloud9. Um das Problem zu beheben, überprüfen Sie die Devfile-Datei aus Ihrem Repository und der entsprechenden Entwicklungsumgebung und erstellen Sie eine neue, um fortzufahren.

Ich habe Probleme in AWS Cloud9

Weitere Probleme finden Sie im Abschnitt zur Fehlerbehebung im [AWS Cloud9 -Benutzerhandbuch](#).

In kann JetBrainsich keine Verbindung zu meinen Entwicklungsumgebungen über herstellen CodeCatalyst

Um das Problem zu beheben, überprüfen Sie, ob Sie nur die neueste Version von JetBrains installiert haben. Wenn Sie mehrere Versionen haben, deinstallieren Sie die älteren Versionen und registrieren Sie Ihren Protokoll-Handler erneut, indem Sie die IDE und den Browser schließen. Öffnen Sie dann den Protokoll-Handler JetBrains und registrieren Sie ihn erneut.

Ich kann nicht AWS Toolkit für meine IDE installieren

Um dieses Problem für VS Code zu beheben, installieren Sie manuell AWS Toolkit for Visual Studio Code von [GitHub](#).

Um dieses Problem für zu beheben JetBrains, installieren Sie manuell AWS Toolkit for JetBrains von [GitHub](#).

In meiner IDE kann ich meine Entwicklungsumgebungen nicht starten

Um dieses Problem für VS Code zu beheben, überprüfen Sie, ob Sie die neueste Version von VS Code und AWS Toolkit for Visual Studio Code installiert haben. Wenn Sie nicht über die neueste

Version verfügen, aktualisieren und starten Sie Ihre Entwicklungsumgebung. Weitere Informationen finden Sie unter [Amazon CodeCatalyst für VS-Code](#).

Um dieses Problem für zu beheben JetBrains, überprüfen Sie, ob Sie die neueste Version von JetBrains und AWS Toolkit for JetBrains installiert haben. Wenn Sie nicht über die neueste Version verfügen, aktualisieren und starten Sie Ihre Entwicklungsumgebung. Weitere Informationen finden Sie unter [Amazon CodeCatalyst für JetBrains](#).

## Behebung von Problemen mit Devfiles

In den folgenden Abschnitten finden Sie Informationen zur Behebung von Problemen im Zusammenhang mit Devfiles in. CodeCatalyst Weitere Informationen zu Devfiles finden Sie unter [Konfiguration Ihrer Entwicklungsumgebung](#).

### Themen

- [Meine Entwicklungsumgebung verwendet die standardmäßige universelle Dev-Datei, obwohl ich ein benutzerdefiniertes Image in einer benutzerdefinierten Dev-Datei implementiert habe](#)
- [Mein Projekt wird nicht in meiner Entwicklungsumgebung mit der standardmäßigen universellen Devdatei erstellt](#)
- [Ich möchte eine Repository-Dev-Datei für eine Entwicklungsumgebung verschieben](#)
- [Ich habe ein Problem beim Starten meiner Dev-Datei](#)
- [Ich bin mir nicht sicher, wie ich den Status meiner Dev-Datei überprüfen soll](#)
- [Meine Dev-Datei ist nicht mit den Tools kompatibel, die im neuesten Image bereitgestellt werden](#)

Meine Entwicklungsumgebung verwendet die standardmäßige universelle Dev-Datei, obwohl ich ein benutzerdefiniertes Image in einer benutzerdefinierten Dev-Datei implementiert habe

Wenn beim Starten einer Entwicklungsumgebung, die eine benutzerdefinierte Entwicklungsdatei verwendet, Fehler CodeCatalyst auftreten, verwendet die Entwicklungsumgebung standardmäßig die standardmäßige universelle Entwicklungsdatei. Um das Problem zu beheben, können Sie den genauen Fehler in den Protokollen unter überprüfen. `/aws/mde/logs/devfile.log` Sie können auch in Ihren Protokollen überprüfen, ob die `postStart` Ausführung erfolgreich war: `/aws/mde/logs/devfileCommand.log`.

## Mein Projekt wird nicht in meiner Entwicklungsumgebung mit der standardmäßigen universellen Devdatei erstellt

Um das Problem zu beheben, stellen Sie sicher, dass Sie keine benutzerdefinierte Entwicklungsdatei verwenden. Wenn Sie keine benutzerdefinierte Entwicklungsdatei verwenden, schauen Sie sich die `devfile.yaml` Datei im Quell-Repository des Projekts an, um Fehler zu finden und zu beheben.

## Ich möchte eine Repository-Dev-Datei für eine Entwicklungsumgebung verschieben

Sie können die Standard-Dev-Datei in `/projects/devfile.yaml` Ihr Quellcode-Repository verschieben. Verwenden Sie den folgenden Befehl, um den Speicherort der Dev-Datei zu aktualisieren: `./aws/mde/mde start --location repository-name/devfile.yaml`

## Ich habe ein Problem beim Starten meiner Dev-Datei

Wenn beim Starten Ihrer Dev-Datei ein Problem auftritt, wechselt sie in den Wiederherstellungsmodus, sodass Sie weiterhin eine Verbindung zu Ihrer Umgebung herstellen und Ihre Dev-Datei reparieren können. Wenn Sie sich im Wiederherstellungsmodus befinden, enthält die Ausführung `./aws/mde/mde status` nicht den Speicherort Ihrer Dev-Datei.

```
{
  "status": "STABLE"
}
```

Sie können den Fehler in den Protokollen unter `überprüfen/./aws/mde/logs`, die Dev-Datei korrigieren und erneut versuchen, es auszuführen `./aws/mde/mde start`.

## Ich bin mir nicht sicher, wie ich den Status meiner Dev-Datei überprüfen soll

Sie können Ihren Devfile-Status überprüfen, indem Sie Folgendes ausführen. `./aws/mde/mde status` Nachdem Sie diesen Befehl ausgeführt haben, wird möglicherweise eine der folgenden Optionen angezeigt:

- `{"status": "STABLE", "location": "devfile.yaml" }`

Dies zeigt an, dass Ihre Dev-Datei korrekt ist.

- `{"status": "STABLE" }`

Dies deutet darauf hin, dass Ihre Dev-Datei nicht gestartet werden konnte und sich im Wiederherstellungsmodus befindet.



Sie können den genauen Fehler in den Protokollen unter `/aws/mde/logs/devfile.log` überprüfen.

Sie können auch in Ihren Protokollen überprüfen, ob die `postStart` Ausführung erfolgreich war: `/aws/mde/logs/devfileCommand.log`.

Weitere Informationen finden Sie unter [Universelle Devfile-Images](#).

## Meine Dev-Datei ist nicht mit den Tools kompatibel, die im neuesten Image bereitgestellt werden

In Ihrer Entwicklungsumgebung `devfile` oder `devfile postStart` kann fehlschlagen, wenn das `latest` Tool nicht über die für ein bestimmtes Projekt erforderlichen Tools verfügt. Gehen Sie wie folgt vor, um das Problem zu beheben:

1. Navigiere zu deiner Dev-Datei.
2. Aktualisieren Sie in Ihrer Dev-Datei stattdessen auf eine detaillierte Image-Version. `latest` Sie könnte wie folgt aussehen:

```
components:
  - container:
    image: public.ecr.aws/amazonlinux/universal-image:1.0
```

3. Erstellen Sie mit der aktualisierten Dev-Datei eine neue Entwicklungsumgebung.

## Behebung von Problemen mit Problemen

Die folgenden Informationen können Ihnen helfen, häufig auftretende Probleme mit Problemen in zu beheben CodeCatalyst.

### Themen

- [Ich kann keinen Beauftragten für mein Problem auswählen](#)

## Ich kann keinen Beauftragten für mein Problem auswählen

Problem: Beim Erstellen eines Problems ist die Liste der Verantwortlichen leer.

Mögliche Lösungen: Die Liste der Beauftragten ist direkt mit den CodeCatalyst Benutzern verknüpft, die als Mitglieder für das Projekt aufgeführt sind. Um zu überprüfen, ob der Zugriff auf das

Benutzerprofil ordnungsgemäß funktioniert, wählen Sie das Profilsymbol und dann Benutzerprofil aus. Wenn die Benutzerprofilinformationen nicht angezeigt werden, überprüfen Sie den Statusbericht auf Vorfälle. Wenn er ausgefüllt wird, reichen Sie ein Serviceticket ein.

## Behebung von Problemen zwischen Amazon CodeCatalyst und den AWS SDKs oder dem AWS CLI

Die folgenden Informationen können Ihnen helfen, häufig auftretende Probleme bei der Arbeit mit CodeCatalyst und den AWS CLI oder den AWS SDKs zu beheben.

### Themen

- [Ich erhalte eine Fehlermeldung, wenn ich über eine Befehlszeile oder ein Terminal `aws codecatalyst` eingebe, dass es sich um eine ungültige Auswahl handelt](#)
- [Ich erhalte einen Anmeldeinformationsfehler, wenn ich `aws codecatalyst` Befehle ausführe](#)

### Ich erhalte eine Fehlermeldung, wenn ich über eine Befehlszeile oder ein Terminal `aws codecatalyst` eingebe, dass es sich um eine ungültige Auswahl handelt

**Problem:** Wenn ich versuche, with AWS CLI zu verwenden CodeCatalyst, werden einer oder mehrere `aws codecatalyst` Befehle nicht als gültig erkannt.

**Lösung:** Die häufigste Ursache für dieses Problem ist, dass Sie eine Version von verwendenAWS CLI, die nicht die neuesten Updates für die neuesten Dienste und Befehle enthält. Aktualisieren Sie Ihre Installation von AWS CLI und versuchen Sie es erneut. Weitere Informationen finden Sie unter [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#).

### Ich erhalte einen Anmeldeinformationsfehler, wenn ich `aws codecatalyst` Befehle ausführe

**Problem:** Wenn ich versuche, with zu verwenden CodeCatalyst, erhalte ich die Meldung `You can configure credentials by running "aws configure". oderUnable to locate authorization token.` AWS CLI

Lösung: Sie müssen ein AWS CLI Profil konfigurieren, um mit CodeCatalyst Befehlen arbeiten zu können. Weitere Informationen finden Sie unter [Einrichtung für die AWS CLI Verwendung von CodeCatalyst](#).

# CodeCatalyst Gesundheitsbericht

Der CodeCatalyst Amazon-Gesundheitsbericht ist ein öffentliches Dashboard, das Benutzern eine aggregierte Liste von up-to-the-minute Benachrichtigungen zur Ressourcenleistung und Verfügbarkeit von Diensten bietet. CodeCatalyst , die weitreichende Auswirkungen haben. Sie können sehen, bei welchen Ressourcen Probleme auftreten und welche sich möglicherweise auf Anwendungen auswirken. CodeCatalyst Auf diese Weise können Sie Ausfälle und andere Ressourcenausfälle systemweit verfolgen. Wenn ein Vorfall auftritt, erscheint eine blaue Anzeige auf dem Symbol für den Gesundheitsbericht. Darüber hinaus CodeCatalyst werden automatisch eine Warnung und eine E-Mail-Benachrichtigung an alle Benutzer mit der Rolle des Space-Administrators im Projekt gesendet, die Details und den Verlauf des Vorfalls nahezu in Echtzeit bereitstellen.

Das Dashboard bietet eine Liste aller aktiven Ereignisse und eine Aufzeichnung von bis zu 100 früheren Vorfällen, die in den letzten 30 Tagen aufgetreten sind. Sie können die Liste der Vorfälle nach dem Datum organisieren, an dem der Vorfall aktualisiert wurde. Sie können die Liste der Vorfälle auch aktualisieren, um aktuelle Updates zu erhalten.

Hier ist ein möglicher Arbeitsablauf für die Verwendung des CodeCatalyst Gesundheitsberichts:

Mateo Jackson ist ein Entwickler im Budding Space mit Space-Administratorrechten. Beim Versuch, eine Pull-Anfrage zu erstellen, erhält er immer wieder eine Fehlermeldung. Er überprüft seine E-Mails und CodeCatalyst stellt fest, dass er eine automatisch generierte E-Mail mit einem Systemvorfall erhalten hat, die einen detaillierten Verlauf des Systemproblems enthält, das seinen Bereich betrifft. Er wählt Update anzeigen und wird zum CodeCatalyst Statusbericht weitergeleitet, in dem er alle vom System gemeldeten Vorfälle einsehen kann. Er wählt den Vorfall aus der Liste aus, um weitere Informationen zu erhalten. Es wird ein geteilter Bildschirm geöffnet, der einen Zeitstempel der letzten Aktualisierung, den Verlauf, die betroffenen Funktionen, die Startzeit und den aktuellen Status des Vorfalls enthält. Er kann auch sehen, dass das Problem weiterhin besteht, aber das Serviceteam hat begonnen, daran zu arbeiten. Jedes Mal, wenn der Verlauf oder der Status des Vorfalls aktualisiert werden, erhält er eine E-Mail. Falls er keinen Zugriff auf seine E-Mail hat, kann er das Glockensymbol im oberen Bereich wählen, um zum CodeCatalyst Gesundheitsbericht zu gelangen.

## CodeCatalyst Konzepte für Gesundheitsberichte

Wenn Sie sich mit den folgenden Konzepten vertraut machen, können Sie den CodeCatalyst Gesundheitsbericht besser verstehen und erfahren, wie Sie damit den Zustand Ihrer Anwendungen, Dienste und Ressourcen verfolgen können.

## Vorfall

Bei dem Vorfall handelt es sich um ein Systemereignis, das sich auf die darin enthaltenen Anwendungen und Ressourcen auswirkt CodeCatalyst. Sie können den Vorfall auswählen, um einen detaillierten Verlauf des Ereignisses anzuzeigen, einschließlich der Zeit, zu der es begann, und ob das Serviceteam an einer Lösung arbeitet.

## Status

Der Status ist der Echtzeitstatus des Vorfalls. Er wird als „In Bearbeitung“ oder „Gelöst“ angezeigt.

## Betroffene Funktionen

Bei den betroffenen Funktionen handelt es sich um die Ressourcen oder Anwendungen, die von dem Vorfall betroffen sind. Ein einzelner Vorfall kann sich auf mehrere Bereiche im System auswirken, darunter Pull-Requests, Probleme, Workflows, Tests, Bereitstellung und Quelle.

## Aktualisiert am

Aktualisiert am gibt einen Zeitstempel der letzten Aktualisierung für den Vorfall an.

# AWS Support für Amazon CodeCatalyst

Wenn Sie einen Bereich erstellen, müssen Sie einen verbinden AWS-Konto und ihn als Rechnungskonto für Ihren Bereich festlegen. Über das Konto, das AWS-Konto Sie als Ihr Rechnungskonto angeben, greifen Sie auch auf Ihren AWS Support Tarif für Amazon CodeCatalyst zu. Wenn Sie Support benötigen, können Sie auf dieser Seite Supportanfragen erstellen. AWS-Konto

CodeCatalyst Benutzer in einem Bereich verwenden die CodeCatalyst Seite AWS Support für Amazon CodeCatalyst , um Supportanfragen zu verwalten. Sie können ein Upgrade auf einen AWS Support Plan wie Business Support oder Enterprise Support durchführen, um CodeCatalyst technische Supportfälle in zu erstellen und zu verwalten CodeCatalyst. Support ist für Supportfälle per Telefon, Internet oder Chat verfügbar.

AWS SupportFür Amazon können nur CodeCatalyst Service- und Ressourcenspezifische Fälle unterstützt werden CodeCatalyst. CodeCatalyst Zu den Ressourcen gehören Ressourcen, die innerhalb von CodeCatalyst und von Benutzern in bereitgestellt werden. Dazu gehören jedoch keine Ressourcen CodeCatalyst, die für andere Dienste AWS oder Dienste von Drittanbietern bereitgestellt werden. Wenn Sie Unterstützung für einen anderen AWS Dienst benötigen, müssen Sie ihn über den öffnenAWS Management Console.

Informationen zum Ändern Ihres Supportplans finden Sie unter [Supportpläne ändern](#).

## Note

Developer Support-Pläne sind nicht für Produktionsumgebungen konzipiert. Wenn ein Space-Abrechnungskonto über einen Developer Support-Plan verfügt, gilt dieser Plan nicht für alle Space-Administratoren und Space-Mitglieder innerhalb vonAWS Support. CodeCatalyst

## Abrechnung AWS Support für Amazon CodeCatalyst

Wenn Sie einen Bereich in erstellen CodeCatalyst, können Benutzer im Bereich Supportanfragen AWS Support für Amazon erstellen und verwalten CodeCatalyst. Sie können zwei Arten von Kundenfällen erstellen:

- Support-Anfragen zu Konto und Abrechnung stehen allen CodeCatalyst Benutzern in diesem Bereich zur Verfügung. Hilfe bei Fragen zur Abrechnung und zum Konto erhalten Sie je nach Ihren Berechtigungen unter CodeCatalyst.
- In Fällen des technischen Supports können Sie sich an einen Mitarbeiter des technischen Supports wenden, der Ihnen bei technischen Problemen und Erweiterungen von Drittanbieteranwendungen weiterhelfen kann. Wenn Sie Basic Support haben, können Sie keinen technischen Supportfall erstellen.

Das als Rechnungskonto für den Raum AWS-Konto angegebene Konto muss über einen Business Support- oder Enterprise Support-Plan verfügen, damit der Raum AWS Support CodeCatalyst für technische Fälle genutzt werden kann.

#### Note

Wenn Ihr Speicherplatz CodeCatalyst von einem Konto aus AWS Support für Amazon genutzt wird, für das es keinen Business Support- oder Enterprise Support-Plan gibt, können Sie ihn trotzdem AWS Support für Konto- und Abrechnungsfälle CodeCatalyst für Amazon nutzen.

Wenn Sie technischen Support benötigen, müssen Sie alle Anfragen über die CodeCatalyst Konsole öffnen. Sie können keine Anfragen für technischen Support CodeCatalyst von [AWS Support](#) in der erstellenAWS Management Console.

#### Note

Anfragen zur Erhöhung des Service-Limits sind AWS Support bei Amazon nicht verfügbar CodeCatalyst. Diese Anfragen können nur vom Root-Benutzer für das Space-Abrechnungskonto in der eingereicht AWS Support Center Console werden.

AWS Support für Amazon CodeCatalyst gelten dieselben Support-Verträge wie AWS Support, allerdings mit den folgenden Überlegungen:

- Schweregrade, Antwortzeiten und SLAs AWS Support gelten für Supportfälle im Abschnitt AWS Support für CodeCatalyst, wie unter [Schweregrad auswählen](#) detailliert beschrieben.

- Space-Administratoren und Space-Mitglieder können die AWS Support APIs, das AWS SDK oder die AWS Support App in Slack nicht verwenden, um Anfragen für zu erstellen. CodeCatalyst CodeCatalyst Support-Fälle können nur von CodeCatalyst eingereicht werden.

**Note**

CodeCatalyst ist nicht vollständig in unsere AWS Incident Detection and Response integriert. AWS Trusted Advisor Prüfen Sie, wie integriert CodeCatalyst ist, um sicherzustellen, dass Ihre Geschäftspraktiken mit der aktuellen Integration übereinstimmen.

Sie müssen ein Benutzer in dem Bereich sein, in dem Sie Support anfordern möchten.

**Note**

Wenn Sie mehr als einen Builder in Ihrem Bereich haben, empfehlen wir Ihnen, einen Business Support- oder Enterprise Support-Plan zu erwerben. Diese Pläne bieten technischen Support für den Raum für bis zu 5.000 Bauarbeiter.

Das als Rechnungskonto für den Raum AWS-Konto angegebene Konto verwendet die `AWSRoleForCodeCatalystSupport` Rolle und die [AmazonCodeCatalystSupportAccess](#) verwaltete Richtlinie. Dadurch können CodeCatalyst Benutzer in einem Bereich auf die CodeCatalyst Seite AWS Support für Amazon zugreifen. Weitere Informationen zu dieser Rolle und Richtlinie finden Sie unter [AmazonCodeCatalystSupportAccess](#). Weitere Überlegungen zur Abrechnung finden Sie unter [Abrechnung verwalten](#) im CodeCatalyst Amazon-Administratorhandbuch.

Hier ist ein möglicher Ablauf für einen Builder, der einen Support-Fall erstellt, in CodeCatalyst:

Mateo Jackson ist Entwickler an einem Projekt in CodeCatalyst. Nachdem Sie sich für Amazon angemeldet haben AWS-Konto, das die Abrechnung AWS Support für Amazon verwaltet, CodeCatalyst und ein Upgrade auf einen Business Support-Plan durchgeführt haben, können alle Entwickler in diesem Bereich Anfragen für technischen Support erstellen. Mateo reicht eine Anfrage an den technischen Support für einen fehlgeschlagenen Workflow in seinem Projekt ein. Mateo verwendet die CodeCatalyst Seite AWS Support für Amazon, um das Formular auszufüllen und einen Fall zu erstellen, wobei er die Workflow-ID und andere Details in der Anfrage angibt. Der Fall wird mit einer Fallnummer erstellt und enthält die Konto-ID des als Rechnungskonto AWS-Konto angegebenen Kontos und ist mit dem Supportplan für den Bereich verknüpft.



Zwar können alle Bauarbeiter Support-Anfragen AWS Support für erstellen CodeCatalyst, es fallen jedoch keine Gebühren für jeden erstellten Fall an. Du kannst auf der Grundlage des AWS Support Premium-Tarifs, den du auf deinem Space-Abrechnungskonto erworben hast, praktisch unbegrenzt Anfragen und Kontakte eröffnen.

#### Note

Das Space-Abrechnungskonto ist AWS-Konto das Konto, für das Ihnen CodeCatalyst Benutzer und Ressourcen in Rechnung gestellt werden. Wenn Sie zwei zusätzliche Dienste bereitgestellt haben AWS-Konten, wenden Sie sich an AWS Support den, um Unterstützung bei Ressourcen zu erhalten, die AWS Management Console für andere Dienste bereitgestellt werden.

Sie können die, für die AWS-Konto Sie bereitgestellt haben, anhand des Workflows identifizieren.

## Deinen Bereich AWS Support für Amazon einrichten CodeCatalyst

AWS Support für Amazon CodeCatalyst verwaltet Supportfälle im Rahmen der AWS Support API-Integration mit CodeCatalyst.

Bei der `AWSRoleForCodeCatalystSupport` Rolle handelt es sich um eine Servicerolle, die für Supportfälle in Ihrem Bereich verwendet wird. Die Rolle muss dem angegebenen Rechnungskonto für den Bereich hinzugefügt werden. Weitere Informationen oder Informationen zum Erstellen der Rolle finden Sie unter [Erstellen der AWSRoleForCodeCatalystSupport Rolle für Ihr Konto und Ihren Bereich](#).

#### Note

Für einen Bereich, der vor dem 20. April 2023 erstellt wurde, müssen Sie die Rolle erstellen, damit der Support für CodeCatalyst für Ihren Bereich funktioniert. Wenn du einen Bereich nach dem 20. April 2023 erstellst, kannst du die Rolle während der Erstellung des Bereichs auf der Seite mit den Abrechnungsdetails erstellen oder indem du auf den Link zum Support-Banner unter klickst CodeCatalyst. CodeCatalyst

## Um den Support für deinen Bereich einzurichten

1. Wenn du einen CodeCatalyst Bereich erstellst, wirst du angewiesen, ein Rechnungskonto zu verbinden. Das angegebene Rechnungskonto für den Speicherplatz wird von abgerechnet. AWS Weitere Informationen zum Erstellen eines Bereichs finden Sie unter [Registrieren Sie sich, um Ihren ersten Bereich und Ihre Entwicklungsrolle zu erstellen](#).
2. Wenn Sie einen CodeCatalyst Bereich erstellen, ist die Option verfügbar, um die `AWSRoleForCodeCatalystSupport` Servicerolle zu erstellen, mit der CodeCatalyst Benutzer auf den Support zugreifen können. Die Rolle verwendet die verwaltete Richtlinie `AmazonCodeCatalystSupportAccess`. Die Rolle muss dem Konto hinzugefügt werden, das als Rechnungskonto für den Bereich AWS-Konto vorgesehen ist. Weitere Informationen zum Erstellen dieser Rolle finden Sie unter [Erstellen der AWSRoleForCodeCatalystSupport Rolle für Ihr Konto und Ihren Bereich](#).
3. Für das angegebene Rechnungskonto für den Bereich wird dem Bereichsadministrator empfohlen, einen Business Support- oder Enterprise Support-Plan für den zu erwerben AWS-Konto. Alle Mitglieder in diesem Bereich können Supportanfragen von Amazon aus AWS Support verwalten CodeCatalyst, und die Supportkanäle werden auf den AWS Support Plan abgestimmt, den Sie gekauft haben und bei dem die Integrationen abgeschlossen sind.
4. Informationen zum Erstellen und Verwalten von Support-Fällen in CodeCatalyst finden Sie unter [Einen CodeCatalyst Supportfall erstellen in CodeCatalyst](#).

## Zugriff auf Support für CodeCatalyst in AWS Management Console

Wenn das für Support aktivierte Rechnungskonto für einen Space getrennt wird, sind AWS Support Fälle, die mit dem vorherigen Space-Abrechnungskonto und dem zugehörigen Supportplan verknüpft sind, AWS Support für Amazon CodeCatalyst nicht mehr sichtbar. Der Root-Benutzer für dieses Rechnungskonto kann alte Fälle von der aus einsehen und lösen AWS Management Console und IAM-Berechtigungen einrichten, damit andere Benutzer alte Fälle einsehen und lösen können. AWS Support Sie können weiterhin die Vorteile Ihres Support-Plans AWS Management Console für alle anderen nutzen AWS-Services und alle Support-Anfragen CodeCatalyst bearbeiten, die zuvor nicht gelöst wurden.

Weitere Informationen finden Sie im [Benutzerhandbuch unter Aktualisierung, Lösung und erneutes Öffnen Ihres Falls](#). AWS Support

Support-Anfragen für allgemeine Anleitungen zu CodeCatalyst können auch in der geöffnet werden AWS Management Console, aber es kann kein technischer Support über diesen Kanal für

erhalten werden. CodeCatalyst Weitere Informationen finden Sie im AWS SupportBenutzerhandbuch unter [Erstellen von Supportanfragen und Fallmanagement](#).

Hier ist ein möglicher Ablauf für einen Benutzer, der einen Supportfall löst, für CodeCatalyst : AWS Management Console

Zwar können alle Bauherren Supportanfragen AWS Support für Amazon erstellen CodeCatalyst, die Support-Anfragen werden jedoch über das Konto abgerechnet, das als Rechnungskonto für den Bereich vorgesehen ist. Mateo Jackson arbeitet als Entwickler an einem Projekt und CodeCatalyst hat einen technischen Support-Fall wegen eines fehlgeschlagenen Workflows in seinem Projekt eröffnet. Das Rechnungskonto für den Bereich, bei dem Sie sich bei Amazon angemeldet CodeCatalyst und einen Business Support-Plan erworben haben, wurde jedoch vom Bereich getrennt. AWS Support Die einzige Möglichkeit für Mateo, die aktuelle Kommunikation einzusehen und Anfragen zu lösen, besteht CodeCatalyst darin, die Fall-ID über das AWS Support Center im zu verwalten. AWS Management Console Zu diesem Zweck erhält Mateo vom Root-Benutzer des vorherigen Space-Abrechnungskontos, das seinem Support-Fall zugeordnet ist, IAM-Berechtigungen und löst den Fall in der Konsole. AWS Support

#### Important

Wenn du das angegebene Abrechnungskonto für deinen Space änderst, ist dein AWS Support Tarif weiterhin bis zum Monatsende verfügbar. AWS Management Console Du musst mit AWS Support dem aktualisierten Rechnungskonto erneut einkaufen, um weiterhin auf deine zuvor erstellten Supportanfragen in CodeCatalyst zugreifen zu können. Wir empfehlen, zu warten, bis Sie alle Ihre Supportfälle gelöst haben, um die Abrechnungskonten AWS Support für Speicherplatz zu ändern, damit der Zugriff auf Ihre Supportanfragen über Amazon nicht beeinträchtigt wird CodeCatalyst.

## Einen CodeCatalyst Supportfall erstellen in CodeCatalyst

Sie können auf der CodeCatalyst Seite AWS Support für Amazon einen Support-Fall erstellen.

Eine AWS Builder-ID kann nur Unterstützung für den Alias erhalten, mit dem sie authentifiziert ist, und nur für Ressourcen, die auf ihren Berechtigungen basieren. Konto- und Abrechnungsoptionen sind für alle Space-Administratoren und Space-Mitglieder verfügbar. Benutzer können jedoch nur Support für Ressourcen erhalten, auf die sie Zugriff haben, CodeCatalyst und nicht in Bezug auf die Verwaltung der Abrechnung für das Konto.

Auf der CodeCatalyst For-Seite für deinen Bereich kannst du ein Konto- und Abrechnungsformular oder eine Anfrage AWS Support für technischen Support für deine CodeCatalyst Ressourcen erstellen.

#### Note

AWS SupportFür Amazon können nur CodeCatalyst Service- und Ressourcenspezifische Fälle unterstützt werden CodeCatalyst. CodeCatalyst Zu den Ressourcen gehören Ressourcen, die innerhalb von CodeCatalyst und von Benutzern in bereitgestellt werden. Dazu gehören jedoch keine Ressourcen CodeCatalyst, die für andere Dienste AWS oder Dienste von Drittanbietern bereitgestellt werden. Wenn Sie Unterstützung für einen anderen AWS Dienst benötigen, müssen Sie ihn über den öffnenAWS Management Console.

Um einen Supportfall zu erstellen in CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

#### Tip

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.


3. Wählen Sie oben auf der Seite das ? Symbol, und wählen Sie dann Support.
4. Wählen Sie Create case (Fall erstellen) aus.
5. Wählen Sie eine der folgenden Optionen:
  - Konto und Abrechnung
  - Technisch

#### Note

AWS SupportBei Amazon CodeCatalyst gilt: Wenn dem Space-Abrechnungskonto ein Business Support- oder Enterprise Support-Plan hinzugefügt wird, steht allen Space-Administratoren und Space-Mitgliedern CodeCatalyst technischer Fallsupport zur Verfügung. Informationen zur Problembeseitigung finden Sie unter [Ich kann keine Anfragen für technischen Support für meine Unterkunft erstellen](#).

AWS SupportDie Pläne erstrecken sich nicht über mehrere Bereiche. Wenn Sie Mitglied mehrerer Spaces sind, muss Ihr Space-Administrator für jeden Space ein AWS Support Premium-Abo erwerben, um technischen Support für alle Bereiche zu erhalten.

6. Wählen Sie den Service, die Kategorie und den Schweregrad aus. Informationen zur Auswahl eines Schweregrads finden Sie unter [Schweregrad auswählen](#).
  - General guidance (Allgemeine Anleitung)
  - System impaired (System beeinträchtigt)
  - Production system impaired (Produktionssystem beeinträchtigt)
  - Production system down (Produktionssystem nicht mehr funktionsfähig)
  - Business-critical system down (Geschäftskritisches System nicht mehr funktionsfähig)
7. Wählen Sie Next step: Additional information (Nächster Schritt: Zusätzliche Informationen).
8. Geben Sie auf der Seite Additional Information (Zusätzliche Informationen) für Subject (Betreff) einen Titel zu Ihrem Problem ein.
9. Befolgen Sie für Description (Beschreibung) die Anweisungen und beschreiben Sie Ihren Fall etwa wie folgt:
  - Informationen zur Problembehandlung, die spezifisch für sind CodeCatalyst, wie Workflow-IDs, Protokolle oder Screenshots
  - Fehlermeldungen, die Sie erhalten haben
  - Von Ihnen befolgte Schritte zur Fehlerbehebung

 Note

Geben Sie keine vertraulichen Informationen in Fallkorrespondenzen weiter, z. B. Anmeldeinformationen, Kreditkarten, signierte URLs oder persönlich identifizierbare Informationen.

10. (Optional) Wählen Sie Attach files (Dateien anhängen), um Ihrem Fall relevante Dateien hinzuzufügen, z. B. Fehlerprotokolle oder Screenshots. Sie können bis zu 3 Dateien anfügen. Jede Datei kann bis zu 5 MB groß sein.
11. Unter Space-Name wird der Name Ihres Spaces angezeigt.

12. Im Feld Builder-Name wird der vollständige Name, der mit Ihrer AWS Builder-ID verknüpft ist, automatisch ausgefüllt.
13. (Optional) Wählen Sie das Projekt unter Projektname aus (falls zutreffend).

 Note

Ihnen werden nur Projekte angezeigt, für die Sie Berechtigungen haben. Wenn Sie Zugriff auf ein anderes Projekt benötigen, bitten Sie Ihren Projektadministrator, Ihnen Zugriff zu gewähren, bevor Sie einen Supportfall erstellen.

14. Wählen Sie Nächster Schritt: Kontaktieren Sie uns.
15. Wählen Sie unter Bevorzugte Kontaktsprache die Standardsprache aus. Derzeit ist nur Englisch verfügbar.
16. Wählen Sie die Option Web, Telefon oder Chat als Kontaktmethode.
17. Überprüfen Sie Ihre Falldetails und wählen Sie dann Senden. Ihre Fall-ID-Nummer und Übersicht werden angezeigt.

Der Support-Fall wird auf Bereichsebene erstellt und ist für alle Mitglieder sichtbar, die Zugriff auf den Bereich und das Projekt (falls ausgewählt) haben, die in Ihrem Support-Fall definiert sind. Es gibt derzeit keine Möglichkeit, eine Support-Anfrage einzelner Benutzer auszulassen.

## Lösung eines Support-Falls in CodeCatalyst

Sie können offene Supportfälle auf der CodeCatalyst Seite AWS Support für Amazon lösen.

Sie müssen in dem Bereich, in dem Sie einen Support-Fall lösen möchten, über die Rolle eines Space-Administrators oder Space-Mitglieds verfügen. Wenn Sie nicht über die Rolle eines Space-Administrators verfügen oder wenn bei der Erstellung des Falls ein Projekt ausgewählt wurde, müssen Sie auch Mitglied des Projekts sein, um den Fall anzeigen und lösen zu können.

Um einen offenen Support-Fall zu lösen in CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

**Tip**

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie oben auf der Seite das? Symbol, und wählen Sie dann AWS Support für Amazon CodeCatalyst.
4. Wählen Sie den Link für den Support-Fall, den Sie verwalten möchten. Wählen Sie Resolve case (Fall lösen).

## Wiedereröffnung eines Support-Falls in CodeCatalyst

Sie können die Option „Einen gelösten Support-Fall erneut öffnen“ auf der CodeCatalyst Seite AWS Support für Amazon verwenden.

**Note**

Sie können Ihren Support-Fall bis zu 14 Tage nach der Behebung Ihres Problems erneut öffnen. Allerdings können Sie einen Fall, der länger als 14 Tage inaktiv war, nicht wieder aufnehmen. Wenn Sie Ihren Fall nicht erneut öffnen können, öffnen Sie einen neuen Fall und geben Sie die vorherige Fallnummer als Referenz an.

Wenn Sie einen bestehenden Fall wieder öffnen, der andere Informationen als Ihr aktuelles Problem enthält, kann der Support-Mitarbeiter Sie auffordern, einen neuen Fall zu erstellen.

Um einen Support-Fall erneut zu öffnen in CodeCatalyst

1. Öffnen Sie die CodeCatalyst Konsole unter <https://codecatalyst.aws/>.
2. Navigiere zu deinem CodeCatalyst Bereich.

**Tip**

Wenn Sie zu mehr als einem Bereich gehören, wählen Sie einen Bereich in der oberen Navigationsleiste aus.

3. Wählen Sie oben auf der Seite das? Symbol, und wählen Sie dann AWS Support für CodeCatalyst.

4. Wählen Sie den Link für den Support-Fall, den Sie verwalten möchten. Wählen Sie Reopen (Wieder aufnehmen) aus. Wählen Sie auf dem Bestätigungsbildschirm OK und dann Senden aus.
5. Füllen Sie die Beschreibung mit den neuesten Informationen zu demselben Problem aus. Geben Sie keine vertraulichen Informationen wie Anmeldeinformationen, Kreditkarten, signierte URLs oder persönlich identifizierbare Informationen weiter.



# Kontingente für CodeCatalyst

In der folgenden Tabelle werden Kontingente und Limits für Amazon beschrieben CodeCatalyst. Zusätzliche Informationen zu bestimmten Aspekten von finden Sie CodeCatalyst in den folgenden Themen:

- [Kontingente für Quell-Repositorys in CodeCatalyst](#)
- [Kontingente für Identität, Erlaubnis und Zugriff in CodeCatalyst](#)
- [Kontingente für Workflows in CodeCatalyst](#)
- [Kontingente für Entwicklungsumgebungen in CodeCatalyst](#)
- [Kontingente für Projekte in CodeCatalyst](#)
- [Kontingente für Blueprints in CodeCatalyst](#)
- [Kontingente für Leerzeichen in CodeCatalyst](#)
- [Kontingente für Ausgaben in CodeCatalyst](#)

|  |       |
|--|-------|
| Maximale Anzahl von Leerzeichen in einem Konto                                     | 5     |
| Maximale Anzahl von Räumen, die ein Benutzer in einem Kalendermonat erstellen kann | 5     |
| Mindestanzahl von AWS-Konten für einen Bereich                                     | 1     |
| Maximale Anzahl von Kontoverbindungen für einen Bereich                            | 5,000 |
| Maximale Anzahl von VPC-Verbindungen für einen Space                               | 100   |
| Maximale Anzahl von Projekten in einem Bereich                                     | 100   |

|  |   |
|--|---|
| <p>Maximale Anzahl von Projekten, zu denen ein Benutzer gehören kann</p> | <p>1.000</p>  |
| <p>Raumbeschreibungen</p>  | <p>Raumbeschreibungen sind optional. Falls angegeben, müssen sie zwischen 0 und 200 Zeichen lang sein. Sie können eine beliebige Kombination aus Buchstaben, Zahlen, Leerzeichen, Punkten, Unterstrichen, Kommas, Bindestrichen und den folgenden Sonderzeichen enthalten:</p> <p>? &amp; \$ % + = / \ ; : \n \t \r</p>   |
| <p>Namen von Leerzeichen</p>   | <p>Raumnamen müssen überall eindeutig sein CodeCatalyst. Sie können Namen gelöschter Leerzeichen nicht wiederverwenden. Raumnamen müssen zwischen 3 und 63 Zeichen lang sein. Sie müssen außerdem mit einem alphanumerischen Zeichen beginnen. Raumnamen können eine beliebige Kombination aus Buchstaben, Zahlen, Punkten, Unterstrichen und Bindestrichen enthalten. Sie dürfen keines der folgenden Zeichen enthalten:</p> <p>! ? @ # \$ % ^ &amp; * ( ) + = { } [ ]   \ &gt; &lt; ~ ` ' " ; :</p> |

# Dokumentenverlauf für Amazon CodeCatalyst

In der folgenden Tabelle werden der Dokumentationsverlauf und die Aktualisierungen der gesamten Dokumentation für beschrieben CodeCatalyst.

| Änderung  | Beschreibung   | Datum         |
|---|--|---------------|
| <a href="#">Neuer Inhalt: Mithilfe von Aufgaben können Probleme in kleinere Ziele unterteilt werden</a> | Es wurden Inhalte hinzugefügt, um das Starten von Aufgaben in Issues zu unterstützen. Aufgaben können zu Ausgaben hinzugefügt werden, um die Bearbeitung dieser Ausgabe weiter aufzuschlüsseln, zu organisieren und nachzuverfolgen.                                   | 4. April 2024 |
| <a href="#">Aktualisierter Inhalt: Einschränkungen von GitHub Aktionen in CodeCatalyst</a>              | Das <a href="#">Einschränkungen von GitHub Aktionen in CodeCatalyst</a> Thema wurde aktualisiert und weist nun darauf hin, dass GitHub Aktionen auf einem älteren Docker-Image der Laufzeitumgebung ausgeführt werden.   | 2. April 2024 |
| <a href="#">Aktualisierter Inhalt: Mit Artefakten arbeiten</a>  | Das <a href="#">Mit Artefakten arbeiten</a> Thema wurde aktualisiert und enthält nun zwei neue Unterthemen: <a href="#">Kann ich Artefakte teilen, ohne sie als Ausgaben und Eingaben anzugeben?</a> und <a href="#">Kann ich Artefakte zwischen Workflows teilen?</a> | 2. April 2024 |

|  |  |               |
|--|--|---------------|
| <a href="#">Neue Inhalte: Referenz zur Aktion „AWS CDK Bereitstellen“</a>                  | Dem wurde eine neue <code>CloudAssemblyRootPath</code> Eigenschaft hinzugefügt. <a href="#">Referenz zur Aktion „AWS CDK Bereitstellen“</a> .  | 1. April 2024 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Docker-Images für die Laufzeitumgebung</a> | Das <a href="#">Arbeiten mit Docker-Images für die Laufzeitumgebung</a> Thema wurde aktualisiert und enthält nun Informationen zum neuen Runtime-Environment-Image vom März 2024.  | 26. März 2024 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Rollen</a>                                 | Konsolidierte Informationen zu Rollenberechtigungen in einer einzigen Tabelle. Die Tabelle befindet sich in einem neuen <a href="#">Für jede Rolle verfügbare Berechtigungen</a> Thema.  | 18. März 2024 |
| <a href="#">Neue Inhalte: Alle Bereiche und Projekte für einen Benutzer anzeigen</a>       | Es wurden Informationen zum Anzeigen eines Eintrags auf der Benutzerhomepage hinzugefügt, in dem jeder CodeCatalyst Bereich oder jedes Projekt für den angemeldeten Benutzer angezeigt wird. CodeCatalyst Siehe <a href="#">Alle Bereiche und Projekte für einen Benutzer anzeigen</a> . | 18. März 2024 |

[Neue Inhalte: Beispiel: Ein Auslöser mit einem Pull und Verzweigungen](#)

Ein Beispiel für einen Pull-Request-Trigger wurde hinzugefügt. Im gesamten [Arbeiten mit Auslösern](#) Thema wurden kleine Korrekturen vorgenommen.

11. März 2024

[Aktualisierter Inhalt: Arbeiten mit Rollen](#)

Die Dokumentation für die Rollen wurde aktualisiert und enthält nun auch Berechtigungen zum Erstellen, Löschen und Anzeigen von Umgebungen.

4. März 2024

[Aktualisierter Inhalt: Tutorial: Verwendung generativer KI-Funktionen](#)

Das Tutorial wurde aktualisiert, um die Änderungen beim Erstellen und Zuweisen von Problemen zu Amazon Q widerzuspiegeln.

4. März 2024

[Neuer Inhalt: Blockieren von GitHub Pull-Request-Zusammenführungen, wenn Workflows fehlschlagen](#)

Es wurden neue Inhalte zum Blockieren von GitHub Pull-Request-Zusammenführungen mit Branch-Schutzregeln hinzugefügt, wenn Workflows fehlschlagen.

4. März 2024

[Neuer Inhalt: Probleme und Komponenten](#)

Es wurden neue Inhalte zur Arbeit mit Problemkomponenten als Entwickler von benutzerdefinierten Blueprints hinzugefügt.

27. Februar 2024

[Aktualisierter Inhalt: Aktionen](#)

Das [CodeCatalyst -Aktionen](#) Thema wurde aktualisiert und enthält nun eine Liste von CodeCatalyst Labs-Aktionen.

21. Februar 2024

[Aktualisierter Inhalt: Arbeiten mit Pull-Requests](#)

Die Dokumentation wurde aktualisiert, um neue Funktionen mit Genehmigungsregeln und übergeordneten Anforderungen für das Zusammenführen von Pull-Requests widerzuspiegeln.

15. Februar 2024

[Aktualisierter Inhalt: Pull-Requests zusammenführen](#)

Es wurde eine Dokumentation für Pull-Requests hinzugefügt, die Informationen über das Überschreiben von Merge-Anforderungen enthält, um einen Pull-Request zusammenzuführen, der noch keine Genehmigungen von den erforderlichen Prüfern erhalten hat oder die Genehmigungsregeln nicht erfüllt hat.

15. Februar 2024

[Neuer Inhalt: Genehmigungsregeln verwalten](#)

Es wurde eine Dokumentation für Pull-Requests hinzugefügt, die Informationen zur Erstellung und Verwaltung von Genehmigungsregeln enthält.

15. Februar 2024

[Aktualisierter Inhalt: Arbeiten mit Rollen](#)

Die Dokumentation für die Rollen wurde aktualisiert und enthält nun auch Berechtigungen für die Arbeit mit Genehmigungsregeln und Pull-Requests.

14. Februar 2024

|  |  |                 |
|--|--|-----------------|
| <a href="#">Aktualisierter Inhalt: Wie behebe ich den Fehler „Workflow-Definition enthält keine Fehler“?</a> | Der <a href="#">Wie behebe ich den Fehler „Workflow-Definition enthält keine Fehler“?</a> Abschnitt wurde aktualisiert und enthält nun weitere Tipps zur Fehlerbehebung.   | 9. Februar 2024 |
| <a href="#">Neue Inhalte: Anzeigen des Workflow-Status</a>   | Es wurde ein Abschnitt hinzugefügt, der den Workflow-Status beschreibt.  | 9. Februar 2024 |
| <a href="#">Inhalt aktualisieren: Kontingente für Workflows in CodeCatalyst</a>                              | Das <a href="#">Kontingente für Workflows in CodeCatalyst</a> Thema wurde mit den Optionen Maximale Anzahl von Aktionen pro Workflow und Maximale Anzahl von Umgebungen, die mit einem AWS-Konto Speicherplatzkontingent verknüpft sind, aktualisiert. | 7. Februar 2024 |
| <a href="#">Aktualisierter Inhalt: Erstellen einer Umgebung</a>  | Der <a href="#">Erstellen einer Umgebung</a> Abschnitt wurde dahingehend aktualisiert, dass Sie maximal eine Kontoverbindung pro Umgebung verwenden können.  | 31. Januar 2024 |
| <a href="#">Neuer Inhalt: Repository für benutzerdefinierte Blueprints GitHub</a>                            | Es wurden neue Inhalte für das GitHub Repository hinzugefügt, die öffentlich verfügbar gemacht werden.   | 10. Januar 2024 |

|  |  |                   |
|--|--|-------------------|
| <a href="#">Aktualisierter Inhalt: Konfiguration von npm mit CodeCatalyst</a>  | Die allgemeinen Konfigurationsanweisungen für die Verwendung von npm mit CodeCatalyst wurden aktualisiert und die Option klarer gestaltet. <code>always-auth=true</code> | 5. Januar 2024    |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Pull-Requests</a>  | Die Dokumentation wurde aktualisiert, um neue Funktionen mit den generativen KI-Funktionen in widerzuspiegeln CodeCatalyst.  | 28. November 2023 |
| <a href="#">Aktualisierter Inhalt: Ein Problem wird erstellt</a>   | Die Dokumentation wurde aktualisiert, um neue Funktionen mit den generativen KI-Funktionen in widerzuspiegeln CodeCatalyst.  | 28. November 2023 |
| <a href="#">Neuer Inhalt: Tutorial: Verwendung generativer KI-Funktionen</a>   | Es wurde ein Tutorial zur Verwendung der generativen KI-Funktionen in Amazon hinzugefügt CodeCatalyst.   | 28. November 2023 |
| <a href="#">Neue Inhalte: Benutzerdefinierte Blueprints und Lebenszyklusmanagement</a>   | Neue Inhalte für die Verwendung von benutzerdefinierten Blueprint- und Lifecycle-Management-Funktionen in Amazon CodeCatalyst hinzugefügt.                               | 8. November 2023  |
| <a href="#">Aktualisierter Inhalt: Tutorial: Erstellen eines Projekts mit dem Blueprint für moderne dreistufige Webanwendungen</a> | Das Tutorial wurde mit Korrekturen und Informationen zur Fehlerbehebung aktualisiert.  | 22. November 2023 |



[Aktualisierter Inhalt: Arbeiten mit Auslösern](#)

Einige Beispiele und Beschreibungen im Zusammenhang mit Pull-Request-Triggern wurden behoben. Ein [Überlegen zu Auslösern bei der Verzweigung](#) Abschnitt wurde hinzugefügt.

22. November 2023

[Neuer Inhalt: Mit SSO anmelden](#)

Es wurden Informationen zur Anmeldung mit Single Sign-On (SSO) und Links zu Informationen zur Einrichtung und Verwaltung eines CodeCatalyst Bereichs hinzugefügt, der den Identitätsverbund unterstützt. Siehe [Einrichten CodeCatalyst](#) und [Mit SSO anmelden](#).

17. November 2023

[Aktualisierter Inhalt: Arbeiten mit Rollen](#)

Die Dokumentation für die Rollen wurde aktualisiert und umfasst nun Berechtigungen für die Arbeit mit Teams, VPC-Verbindungen, Single Sign-On und Maschinenressourcen.

16. November 2023

[Aktualisierter Inhalt: Arbeiten mit Pull-Requests](#)

Die Dokumentation wurde aktualisiert, um Änderungen in der Darstellung von Änderungen für einen Pull-Request widerzuspiegeln.

16. November 2023

|   |   |                   |
|---|---|-------------------|
| <a href="#">Aktualisierter Inhalt: Kontingente für CodeCatalyst</a>   | Das <a href="#">Kontingente für CodeCatalyst</a> Thema mit der Option Maximale Anzahl von VPC-Verbindungen für ein Speicherkontingent wurde aktualisiert.         | 16. November 2023 |
| <a href="#">Neuer Inhalt: Verwenden von Entwicklungsumgebungen mit einer VPC-Verbindung</a>                                 | Dokumentation für die Verwendung von Entwicklungsumgebungen mit einer VPC-Verbindung hinzugefügt.   | 16. November 2023 |
| <a href="#">Neuer Inhalt: Teams für einen Bereich und für CodeCatalyst Projekte verwalten</a>                               | Es wurden Informationen zur Verwendung von Teams mit Bereichen hinzugefügt. Siehe <a href="#">Teams managen</a> und <a href="#">Teams für Projekte leiten</a> .   | 16. November 2023 |
| <a href="#">Neuer Inhalt: Verwaltung von Maschinenressourcen für Blueprints und Workflows in einem Bereich</a>              | Es wurden Informationen zur Verwendung von Maschinenressourcen mit Räumen hinzugefügt. Siehe <a href="#">Verwaltung von Maschinenressourcen</a> .                 | 16. November 2023 |
| <a href="#">Neuer Inhalt: Verwaltung von Maschinenressourcen für Blueprints und Workflows in einem Projekt CodeCatalyst</a> | Es wurden Informationen zur Verwendung von Maschinenressourcen mit CodeCatalyst Projekten hinzugefügt. Siehe <a href="#">Verwaltung von Maschinenressourcen</a> . | 16. November 2023 |
| <a href="#">Neuer Inhalt: Eine VPC-Verbindung mit einer Umgebung verknüpfen</a>   | Es wurde eine Dokumentation für die Zuordnung einer VPC-Verbindung zu einer Umgebung hinzugefügt, die in einem Workflow verwendet werden kann.                    | 16. November 2023 |

|   |  |                   |
|---|--|-------------------|
| <a href="#">Neuer Inhalt</a>  | Erstveröffentlichung des <a href="#">Amazon CodeCatalyst Administrator Guide</a> .   | 16. November 2023 |
| <a href="#">Neuer Inhalt: Referenz zur Aktion „AWS CDK Bereitstellen“</a>   | Dem <a href="#">Referenz zur Aktion „AWS CDK Bereitstellen“</a> und dem wurde eine neue <code>CdkCliVersion</code> Eigenschaft hinzugefügt <a href="#">Referenz zur Aktion „AWS CDK bootstrap“</a> . | 14. November 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Rollen</a>  | Die Dokumentation für die Rollen wurde aktualisiert und enthält nun auch Berechtigungen für die Arbeit mit Branch-Regeln.  | 13. November 2023 |
| <a href="#">Aktualisierter Inhalt: Behebung von Problemen mit Quell-Repositorys, Workflows und Entwicklungsumgebungen</a> | Die Themen zur Problembehandlung wurden aktualisiert und enthalten nun Informationen zum Arbeiten mit Branch-Regeln.   | 13. November 2023 |
| <a href="#">Aktualisierter Inhalt: Aktionsreferenz zum Erstellen und Testen</a>   | Die Dokumentation für die <code>Environment</code> Immobilie wurde aktualisiert. Es ist jetzt ein optionales Feld für Build- und Testaktionen.   | 13. November 2023 |
| <a href="#">Neuer Inhalt: Filialregeln verwalten</a>  | Es wurde eine Dokumentation für Branches hinzugefügt, die Informationen zum Anzeigen von Regeln für Branches in einem Quell-Repository sowie zum Erstellen und Verwalten von Branch-Regeln enthält.  | 13. November 2023 |

---

|  |   |                   |
|--|---|-------------------|
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Pull-Requests</a>                            | Die Dokumentation wurde aktualisiert, um Änderungen in der Darstellung von Informationen zu Pull-Requests widerzuspiegeln.  | 10. November 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Datei-Caching</a>                            | Die Dokumentation wurde aktualisiert und enthält nun Einschränkungen beim Zwischenspeichern von Dateien.  | 10. November 2023 |
| <a href="#">Aktualisierter Inhalt: Tutorial: Bereitstellen einer Anwendung in Amazon EKS</a> | Die Dokumentation wurde aktualisiert, um den EKS App Deployment Blueprint zu erwähnen.  | 9. November 2023  |
| <a href="#">Neuer Inhalt: Pakete in CodeCatalyst</a>   | Dokumentation zur Verwendung von Paketen in hinzugefügt CodeCatalyst.   | 1. November 2023  |
| <a href="#">Neuer und aktualisierter Inhalt: Arbeiten mit Rollen</a>                         | Die Dokumentation für vier neue Rollen in den CodeCatalyst Kategorien Hauptbenutzer, Eingeschränkter Zugriff, Überprüfer und Schreibgeschützt wurde aktualisiert. | 1. November 2023  |

[Inhalt aktualisiert: Mit GitHub Aktions-Ausgabeparametern arbeiten](#)

Die Beispiele wurden aktualisiert, sodass die GITHUB\_OUTPUT Umgebungsdatei anstelle des set-output Befehls verwendet wird. Die Verwendung von Umgebungsdateien GitHub ist die empfohlene Methode zur Einstellung von Ausgabeparametern.

24. Oktober 2023

[Neuer Inhalt: Arbeiten mit Auslösern](#)

Dokumentation für Zeitplan-Trigger hinzugefügt.

16. Oktober 2023

[Aktualisierter Inhalt: Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#)

Informationen zur Verwendung der CodeCatalystWorkflowDevelopmentRole-*spaceName* Rolle wurden zu den [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#) Themen [Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“](#) und hinzugefügt.

22. September 2023

[Aktualisierter Inhalt: Neuer Rollenname und neue Richtlinie für die CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Rolle](#)

Die Schritte und Rollenbeschreibungen für die Änderung des Namens der Entwicklungsrolle wurden auf aktualisiert CodeCatalystWorkflowDevelopmentRole-*spaceName* . Die Entwicklungsrolle verwendet jetzt die AdministratorAccess AWS verwaltete Richtlinie. Siehe [Grundlegendes zur CodeCatalystWorkflowDevelopmentRole-\*spaceName\* Servicerolle](#) und [Registrieren Sie sich, um Ihren ersten Bereich und Ihre Entwicklungsrolle zu erstellen](#).

20. September 2023

[Aktualisierter Inhalt: Arbeiten mit Variablen](#)

Zwei neue Konzepte wurden eingeführt: benutzerdefinierte Variablen und vordefinierte Variablen. Diese Konzepte sollten das Lesen und Verstehen des [Arbeiten mit Variablen](#) Abschnitts erleichtern.

19. September 2023

[Aktualisierter Inhalt: Mit Commits arbeiten](#)

Die Dokumentation wurde aktualisiert, um der Änderung der angezeigten Informationen Rechnung zu tragen und weitere Informationen zur Anzeige von Commits mit mehreren übergeordneten Elementen bereitzustellen.

07. September 2023

|  |  |                   |
|--|--|-------------------|
| <a href="#">Neuer Inhalt: Arbeiten mit Auslösern</a>   | Das folgende Beispiel wurde dem <a href="#">Arbeiten mit Auslösern</a> Thema hinzugefügt: <a href="#">Beispiel: Ein einfacher „Push to Main“-Auslöser</a>                                  | 6. September 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Pull-Requests</a>                                  | Die Dokumentation wurde aktualisiert, um die Änderung der Anzeigereihenfolge für den Quell-Branch und den Ziel-Branch beim Erstellen eines Pull-Requests widerzuspiegeln.                  | 30. August 2023   |
| <a href="#">Neuer Inhalt: Den Standardzweig anzeigen und ändern</a>                                | Es wurde eine Dokumentation für Branches hinzugefügt, die Informationen zum Anzeigen und Ändern des Standard-Branches für ein Quell-Repository enthält.                                    | 30. August 2023   |
| <a href="#">Aktualisierter Inhalt: Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“</a> | Der <a href="#">Manifests</a> Eigenschaftsbeschreibung im wurde ein Hinweis zu Helm und Kustomize hinzugefügt. <a href="#">Referenz zur Aktion „Bereitstellung für Kubernetes-Cluster“</a> | 15. August 2023   |
| <a href="#">Neuer Inhalt: Verwaltung von Problemanhängen</a>                                       | Es wurde eine Dokumentation für die Arbeit mit und die Verwaltung von Anhängen zu Problemen hinzugefügt.   | 15. August 2023   |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Auslösern</a>                                      | Die Dokumentation zu Workflow-Triggern wurde verbessert und erweitert.   | 11. August 2023   |

[Neuer Inhalt: Problembehandlung bei Rollenberechtigungen](#)

Es wurden Informationen zur Aktualisierung der Rollenberechtigungen für die Ausführung eines Workflows hinzugefügt, für den Zugriff auf Amazon erforderlich ist CodeGuru. Siehe [Moderner dreistufiger Blueprint-Workflow für Webanwendungen OnPullRequests schlägt mit einem Berechtigungsfehler für Amazon fehl CodeGuru](#).

11. August 2023

[Neuer Inhalt: Wie behebe ich die Meldung „Der Workflow ist inaktiv“?](#)

Das folgende Thema zur Fehlerbehebung wurde hinzugefügt: [Wie behebe ich die Meldung „Der Workflow ist inaktiv“?](#)

11. August 2023

[Neuer Inhalt: Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#)

Dokumentation für die Cluster-Aktion „Auf Kubernetes bereitstellen“ wurde hinzugefügt. Weitere Informationen finden Sie unter [Aktion „Auf Kubernetes-Cluster bereitstellen“ hinzufügen](#) und [Tutorial: Bereitstellen einer Anwendung in Amazon EKS](#).

27. Juli 2023



[Updates dafür, wie Verwaltungseignisse für einen Space protokolliert werden CodeCatalyst](#)

Es wurden Informationen darüber hinzugefügt, wie Verwaltungseignisse für bestimmte Aktionen in einem CodeCatalyst Bereich mit protokolliert AWS CloudTrail werden. Es wurden Informationen darüber hinzugefügt, wie alle Ereignisse in einem Bereich mit dem list-event-logs Befehl angezeigt werden können. Siehe [Überwachung bei Amazon CodeCatalyst](#).

20. Juli 2023

[Aktualisierter Inhalt: Arbeiten mit Auslösern](#)

Die Dokumentation wurde aktualisiert, um darauf hinzuweisen, dass Pull-Request-Trigger jetzt mit GitHub Quell-Repositorys unterstützt werden. Bisher wurden Pull-Request-Trigger nur mit CodeCatalyst Quell-Repositorys unterstützt.

14. Juli 2023

[Aktualisierter Inhalt: Referenz zur Workflow-Definition](#)

Ein Formatierungsfehler im Compute Codeblock wurde behoben.

27. Juni 2023

[Aktualisierter Inhalt: Kontingente für Workflows in CodeCatalyst](#)

Das [Kontingente für Workflows in CodeCatalyst](#) Thema wurde mit der Option Maximaler Zeitraum, für den eine Aktion ein Kontingent ausgeführt werden kann, aktualisiert.

27. Juni 2023

---

|   |  |               |
|---|--|---------------|
| <a href="#">Aktualisierter Inhalt: Datenschutz</a>                                | Die Dokumentation wurde aktualisiert und enthält nun zusätzliche Informationen zur Datenreplikation.   | 26. Juni 2023 |
| <a href="#">Neuer Inhalt: Mit Action-Versionen arbeiten</a>                       | Ein <a href="#">Mit Action-Versionen arbeiten</a> Thema wurde hinzugefügt.   | 21. Juni 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Umgebungen</a>                    | Der <a href="#">Welche Aktionen unterstützen Umgebungen?</a> Abschnitt wurde geklärt.  | 14. Juni 2023 |
| <a href="#">Inhalt aktualisiert: Dokumentation zu neu organisierten Problemen</a> | Der Großteil der Problemdokumentation wurde neu organisiert, um sie besser an die gesamte Dokumentation und die Benutzerabläufe anzupassen.        | 31. Mai 2023  |
| <a href="#">Aktualisierter Inhalt: Issues View Switcher</a>                       | Verschiedene Benutzerabläufe wurden aktualisiert, sodass sie dem aktualisierten Problemanichts-Switcher entsprechen.                               | 31. Mai 2023  |
| <a href="#">Aktualisierter Inhalt: Benachrichtigungen verwalten</a>               | Die Dokumentation für Benachrichtigungen wurde aktualisiert und enthält nun Informationen zur Konfiguration persönlicher Slack-Benachrichtigungen. | 30. Mai 2023  |
| <a href="#">Aktualisierter Inhalt: Benachrichtigungen verwalten</a>               | Die Dokumentation für Benachrichtigungen wurde aktualisiert und enthält nun Informationen zur Konfiguration persönlicher Slack-Benachrichtigungen. | 30. Mai 2023  |

---

|  |  |              |
|--|--|--------------|
| <a href="#">Neuer Inhalt: CodeCatalyst Vertrauensmodell</a>                      | Es wurde ein neues Thema mit Informationen zum Vertrauensmodell hinzugefügt, das es CodeCatalyst ermöglicht, die Servicerolle im verbundenen Bereich zu übernehmen AWS-Konto. Es wurde ein neuer Abschnitt über die definierten Dienstprinzipale für CodeCatalyst hinzugefügt. Siehe <a href="#">CodeCatalyst Vertrauensmodell</a> . | 20. Mai 2023 |
| <a href="#">Aktualisierter Inhalt: Mit Quellen arbeiten</a>                      | Vereinfacht die Anweisungen in <a href="#">Verweisen auf Dateien in einem Quell-Repository</a> .   | 10. Mai 2023 |
| <a href="#">Aktualisierter Inhalt: Kontingente für Workflows in CodeCatalyst</a> | Das <a href="#">Kontingente für Workflows in CodeCatalyst</a> Thema wurde um das Kontingent „Maximale Länge eines Werts für eine Ausgabevariable“ aktualisiert.  | 10. Mai 2023 |
| <a href="#">Neuer Inhalt: Mit Artefakten arbeiten</a>                            | Zwei Beispiele wurden hinzugefügt: <a href="#">Beispiel: Referenzieren einer Datei in einem einzigen Artefakt</a> und <a href="#">Beispiel: Verweisen auf eine Datei in einem Artefakt, wenn a vorhanden ist WorkflowSource</a> .  | 10. Mai 2023 |

|   |  |                |
|---|--|----------------|
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Pull-Requests</a>   | Die Dokumentation für Pull-Requests wurde aktualisiert und enthält nun Informationen zur Konfiguration von E-Mail-Einstellungen für Pull-Request-Ereignisse.   | 21. April 2023 |
| <a href="#">Aktualisierter Inhalt: Benachrichtigungen verwalten</a> | Die Dokumentation für Benachrichtigungen wurde aktualisiert und enthält nun Informationen zur Konfiguration der E-Mail-Einstellungen für Pull-Request-Ereignisse.  | 21. April 2023 |
| <a href="#">Aktualisierungen der verwalteten Richtlinien</a>        | Die <a href="#">AWS verwaltete Richtlinie: AmazonCodeCatalyst FullAccessAWS verwaltete Richtlinie: AmazonCodeCatalystReadOnlyAccess</a> , und <a href="#">AWS verwaltete Richtlinie: AmazonCodeCatalyst SupportAccess</a> verwalteten Richtlinien wurden hinzugefügt. Siehe <a href="#">CodeCatalyst -Aktualisierungen für - AWSverwaltete Richtlinien</a> . | 20. April 2023 |
| <a href="#">Neuer Inhalt: Ein Bereitstellungsziel entfernen</a>     | Ein <a href="#">Ein Bereitstellungsziel entfernen</a> Thema wurde hinzugefügt.   | 20. April 2023 |
| <a href="#">Neuer Inhalt: Aktionstypen</a>                          | Ein <a href="#">CodeCatalyst Labs-Aktionen</a> Thema wurde hinzugefügt.  | 20. April 2023 |

|   |  |                |
|---|--|----------------|
| <a href="#">Updates für die Verwaltung eines Benutzers mit der Rolle Space-Administrator in einem Bereich</a> | Es wurden Informationen zum Entfernen oder Ändern der Rolle für einen Benutzer mit der Rolle Space-Administrator in einem Bereich hinzugefügt. Siehe <a href="#">Entfernen oder Ändern der Rolle für einen Benutzer mit der Space-Administrato rolle</a> . | 19. April 2023 |
| <a href="#">Updates für die Verwaltung von Entwicklungsumgebungen</a>   | Es wurden Informationen zur Verwaltung von Entwicklungsumgebungen als Space-Administrator hinzugefügt. Siehe <a href="#">Verwaltung von Entwicklungsumgebungen für einen Bereich</a> .   | 19. April 2023 |
| <a href="#">Aktualisierter Inhalt: Probleme finden und anzeigen</a>   | Das <a href="#">Probleme finden und anzeigen</a> Thema und die Unterthemen wurden neu organisiert.   | 19. April 2023 |
| <a href="#">Inhalt aktualisiert: Mit Computearbeiten</a>  | Unterstützung für die Arm64-Architektur auf Amazon Linux 2 hinzugefügt.  | 19. April 2023 |
| <a href="#">Neuer Inhalt: Probleme innerhalb von Gruppen verschieben</a>                                      | Es wurde <a href="#">eine Dokumentation zum Verschieben von Problemen innerhalb von Gruppen in</a> den Ansichten Forum und Alle Probleme hinzugefügt.  | 19. April 2023 |

|   |  |                |
|---|--|----------------|
| <a href="#">Aktualisierter Inhalt: Kontingente für Workflows in CodeCatalyst</a>  | Das <a href="#">Kontingente für Workflows in CodeCatalyst</a> Thema wurde mit fehlenden Kontingenten aktualisiert und das Kontingent Maximale Gesamtgröße der Ausgabevariablen einer einzelnen Aktion auf 120 KB (von 2 KB) aktualisiert.  | 18. April 2023 |
| <a href="#">Neuer Inhalt: Den Quellcode einer Aktion anzeigen</a>   | Ein <a href="#">Den Quellcode einer Aktion anzeigen</a> Thema wurde hinzugefügt.   | 18. April 2023 |
| <a href="#">Neuer Inhalt: Testfälle erneut versuchen</a>  | Ein <a href="#">Testfälle erneut versuchen</a> Thema wurde hinzugefügt.  | 11. April 2023 |
| <a href="#">Neuer Inhalt: Anhalten einer Workflow-Ausführung</a>  | Ein <a href="#">Anhalten einer Workflow-Ausführung</a> Thema wurde hinzugefügt.  | 10. April 2023 |
| <a href="#">Neuer Inhalt: Abschnitte zum Markieren von Ressourcen für Kontoverbindungen zwischen Amazon AWS und Amazon hinzugefügt CodeCatalyst</a> | Es wurden Informationen zum Markieren von Kontoverbindungsressourcen und zum Verwalten von IAM-Richtlinien für Verbindungsressourcen hinzugefügt. Siehe <a href="#">Verwenden von Tags zur Steuerung des Zugriffs auf Kontoverbindungsressourcen</a> und <a href="#">CodeCatalyst -Berechtigungsreferenz</a> . | 06. April 2023 |
| <a href="#">Neuer Inhalt: Aktionstypen</a>  | Ein <a href="#">Aktionstypen</a> Thema wurde hinzugefügt.  | 06. April 2023 |

|   |  |               |
|---|--|---------------|
| <a href="#">Aktualisierter Inhalt: Referenz zur Aktion „Stack bereitstellenAWS CloudFormation“</a>          | Die <code>parameter-override</code> s Objektbeschreibung wurde aktualisiert. Es unterstützt jetzt JSON-Dateien.  | 5. April 2023 |
| <a href="#">Neuer Inhalt: Erstellen eines Projekts CodeCatalyst mit einem verknüpften GitHub Repository</a> | Es wurde ein neuer Abschnitt <a href="#">Ein Projekt mit einem verknüpften GitHub Repository erstellen</a> mit dem <a href="#">Ein Projekt in Amazon erstellen CodeCatalyst</a> Titel hinzugefügt, der Anweisungen zum Erstellen eines Projekts enthält, das auf Ihr GitHub Repository verweist. | 5. April 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Benachrichtigungen</a>                                      | Die Dokumentation für Benachrichtigungen wurde aktualisiert und enthält nun Informationen zur Konfiguration von E-Mails über Projektereignisse.  | 31. März 2023 |
| <a href="#">Neuer Inhalt</a>  | Erste Veröffentlichung des <a href="#">Amazon CodeCatalyst Action Development Kit-Leitfadens</a> .   | 31. März 2023 |
| <a href="#">Aktualisierter Inhalt: Der Bereich Spaces in Amazon wurde neu strukturiert CodeCatalyst</a>     | Der Bereich Spaces wurde aktualisiert, indem Landingpages entfernt und Themen konsolidiert wurden.   | 29. März 2023 |

|   |  |               |
|---|--|---------------|
| <a href="#">Inhalt aktualisiert: Tutorial: Bereitstellen einer Anwendung in Amazon ECS</a>            | Es wurde geändert <a href="#">Schritt 1: Richten Sie einen AWS Benutzer ein und AWS CloudShell</a> , um zu beschreiben, wie ein Benutzer AWS IAM Identity Center anstelle von erstellt wird AWS Identity and Access Management. Das Erstellen von IAM-Benutzern wird nicht mehr empfohlen. | 23. März 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Rollen</a>  | Die Dokumentation für die Rollen Space-Administrator, Projektadministrator und Mitwirkender wurde aktualisiert und enthält nun auch Berechtigungen zum Verknüpfen von Issues mit Pull Requests.  | 13. März 2023 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Pull Requests</a>                                     | Die Dokumentation für Pull-Requests wurde aktualisiert und enthält nun Informationen zum Verknüpfen von Problemen mit Pull-Requests.   | 13. März 2023 |
| <a href="#">Aktualisierter Inhalt: Mit Problemen arbeiten</a>   | Die Dokumentation für Probleme wurde aktualisiert und enthält nun Informationen zum Verknüpfen von Problemen mit Pull-Requests.  | 13. März 2023 |
| <a href="#">Neue Inhalte: Anzeigen des Status und der Details aller Ausführungen in Ihrem Projekt</a> | Es wurde ein Abschnitt hinzugefügt, der die neue aggregierte Workflow-Ausführungsseite beschreibt.   | 08. März 2023 |



---

|  |  |                  |
|--|--|------------------|
| <a href="#">Neuer Inhalt: Wie behebe ich den Fehler „Workflow-Definition enthält <i>keine</i> Fehler“?</a> | Es wurde ein Abschnitt zur Behebung von Fehlern mit dem Hinweis „Die Workflow-Definition enthält Fehler“ hinzugefügt.  | 7. März 2023     |
| <a href="#">Aktualisierter Inhalt: Einen Workflow erstellen, bearbeiten und löschen</a>                    | Die Anweisungen wurden aktualisiert, um sie an die neue Benutzeroberfläche anzupassen.   | 03. März 2023    |
| <a href="#">Neuer Inhalt: Benutzen universal-test-runner</a>   | Ein <a href="#">Benutzen universal-test-runner</a> Thema wurde hinzugefügt.  | 03. März 2023    |
| <a href="#">Aktualisierter Inhalt: Erstellen, Testen und Bereitstellen mit Workflows in CodeCatalyst</a>   | Verschiedene Abschnitte wurden aktualisiert, um die neuen Quell-Repository-, Branch- und Workflow-Namensfilter auf der Workflow-Übersichtsseite widerzuspiegeln. | 2. März 2023     |
| <a href="#">Neuer Inhalt: Codequalität und Bereitstellungsstatus nach Commit in anzeigen CodeCatalyst</a>  | Es wurde ein Abschnitt zur Anzeige der Codequalität und des Bereitstellungsstatus nach Commit hinzugefügt.   | 27. Februar 2023 |
| <a href="#">Neuer Inhalt: Variablen BranchName "" und CommitId ""</a>                                      | Eine neue BranchName vordefinierte Variable wurde hinzugefügt.   | 16. Februar 2023 |

---

|   |  |                   |
|---|--|-------------------|
| <a href="#">Aktualisierter Inhalt: Space-Mitglieder in Amazon verwalten CodeCatalyst</a>                              | Die Informationen zum Ändern von Mitgliederrollen, zum Einladen von Mitgliedern und zum Entfernen von Mitgliedern wurden in zwei neuen Tabellen aktualisiert, basierend auf der dem Benutzer zugewiesenen Rolle in CodeCatalyst. | 15. Februar 2023  |
| <a href="#">Aktualisierter Inhalt: Schritte für die PAT-Verwaltung in der CodeCatalyst Amazon-Konsole hinzugefügt</a> | Es wurden Schritte zum Anzeigen, Erstellen und Löschen von PATs in der Konsole hinzugefügt.  | 15. Februar 2023  |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Docker-Images für die Laufzeitumgebung</a>                            | Der Tabelle mit den Versionen der Standard-Image-Tools wurden weitere Tools hinzugefügt.   | 10. Januar 2023   |
| <a href="#">Aktualisierter Inhalt: Mit Artefakten arbeiten</a>  | Ein Artefaktpfad wurde behoben.  | 3. Januar 2023    |
| <a href="#">Inhalt aktualisiert: Mit Quellen arbeiten</a>   | Ein Quellpfad wurde korrigiert.  | 3. Januar 2023    |
| <a href="#">Aktualisierter Inhalt: Referenz zur Aktion „GitHub Aktionen“</a>  | Das Code-Snippet im Abschnitt wurde behoben. Steps   | 3. Januar 2023    |
| <a href="#">Aktualisierter Inhalt: Aktualisierung einer Pull-Anfrage</a>  | Die Dokumentation wurde aktualisiert und enthält nun Informationen zur Aktualisierung erforderlicher oder optionaler Reviewer für einen Pull-Request.  | 23. Dezember 2022 |

|  |   |                   |
|--|---|-------------------|
| <a href="#">Neuer Inhalt: Arbeiten mit Datei-Caching</a>                           | Es wurde eine Seite für das Zwischenspeichern von Dateien in einem Workflow hinzugefügt.                    | 20. Dezember 2022 |
| <a href="#">Aktualisierter Inhalt: Arbeiten mit Pull-Requests</a>                  | Die Dokumentation für Pull-Requests wurde aktualisiert und enthält nun Informationen zu Benachrichtigungen. | 16. Dezember 2022 |
| <a href="#">Neuer Inhalt: Referenz zur Aktion „AWS CDK Bereitstellen“</a>          | Eine neue CdkRootPath Eigenschaft wurde hinzugefügt.  | 16. Dezember 2022 |
| <a href="#">Neuer Inhalt: Rechenleistung für mehrere Aktionen gemeinsam nutzen</a> | Ein <a href="#">Rechenleistung für mehrere Aktionen gemeinsam nutzen</a> Thema wurde hinzugefügt.           | 14. Dezember 2022 |
| <a href="#">Aktualisierter Inhalt: Mit Artefakten arbeiten</a>                     | Es wurden Beispiele behoben, die zeigen, wie Eingabeartefakte spezifiziert werden.                          | 13. Dezember 2022 |
| <a href="#">Neuer Inhalt: Referenz zur Aktion „GitHub Aktionen“</a>                | Es wurde eine spezielle Referenzseite für die Aktion GitHub Aktionen hinzugefügt.                           | 13. Dezember 2022 |
| <a href="#">Aktualisierter Inhalt: Kontingente für Projekte in CodeCatalyst</a>    | Die Dokumentation wurde mit maximal 100 Projekten in einem Bereich aktualisiert.                            | 2. Dezember 2022  |
| <a href="#">Neuer Inhalt</a>   | Erstveröffentlichung des CodeCatalyst Amazon-Benutzerhandbuchs.   | 01. Dezember 2022 |

# AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.