User Guide

AWS Batch

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

AWS Batch: User Guide

Copyright © 2025 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 Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS Batch? ... 1
Sind Sie ein AWS Batch Erstanwender? ... 3
Zugehörige Services ... 3
Zugreifen AWS Batch ... 3
Bestandteile von AWS Batch ... 4
Datenverarbeitungsumgebung ... 4
Warteschlangen für Job ... 5
Berufsdefinitionen ... 5
Aufträge ... 5
Richtlinie zur Terminplanung ... 6
Verbrauchbare Ressourcen ... 6
Einrichten AWS Batch ... 7
IAM-Konto und Administratorbenutzer erstellen ... 7
Melde dich an für ein AWS-Konto ... 7
Erstellen eines Benutzers mit Administratorzugriff ... 8
Erstellen Sie IAM-Rollen ... 10
Erstellen eines Schlüsselpaares ... 10
Erstellen einer VPC ... 12
Eine Sicherheitsgruppe erstellen ... 13
Installiere das AWS CLI ... 15
Tutorials für den Einstieg ... 16
Erste Schritte mit Amazon EC2 mithilfe des Wizards ... 16
Übersicht ... 16
Voraussetzungen ... 17
Schritt 1: Erstellen Sie eine Rechenumgebung ... 18
Schritt 2: Erstellen Sie eine Job-Warteschlange ... 18
Schritt 3: Erstellen Sie eine Jobdefinition ... 19
Schritt 4: Erstellen Sie einen Job ... 19
Schritt 5: Überprüfen und Erstellen ... 20
Schritt 6: Sehen Sie sich die Ausgabe des Jobs an ... 20
Schritt 7: Bereinigen Sie Ihre Tutorial-Ressourcen ... 20
Weitere Ressourcen ... 21
Erste Schritte mit der Fargate-Orchestrierung mithilfe des Wizards ... 21
Übersicht ... 21

Voraussetzungen ... 22
Schritt 1: Erstellen Sie eine Rechenumgebung ... 23
Schritt 2: Erstellen Sie eine Job-Warteschlange ... 23
Schritt 3: Erstellen Sie eine Jobdefinition ... 24
Schritt 4: Erstellen Sie einen Job ... 25
Schritt 5: Überprüfen und Erstellen ... 25
Schritt 6: Sehen Sie sich die Ausgabe des Jobs an ... 25
Schritt 7: Bereinigen Sie Ihre Tutorial-Ressourcen ... 26
Weitere Ressourcen ... 26
Erste Schritte mit AWS Batch und Fargate mit dem AWS CLI ... 27
Voraussetzungen ... 27
Erstellen Sie eine IAM-Ausführungsrolle ... 28
Erstellen Sie eine Rechenumgebung ... 30
Erstellen Sie eine Job-Warteschlange ... 31
Erstellen Sie eine Auftragsdefinition ... 32
Einen Job einreichen und überwachen ... 33
Job-Ausgabe anzeigen ... 34
Bereinigen von -Ressourcen ... 35
Wir gehen zur Produktion ... 36
Nächste Schritte ... 37
Erste Schritte mit Amazon EKS ... 37
Übersicht ... 39
Voraussetzungen ... 40
Schritt 1: Erstellen Sie Ihren Amazon EKS-Cluster für AWS Batch ... 41
Schritt 2: Bereiten Sie Ihren Amazon EKS-Cluster vor für AWS Batch ... 41
Schritt 3: Erstellen Sie eine Amazon EKS-Rechenumgebung ... 45
Schritt 4: Erstellen Sie eine Job-Warteschlange und hängen Sie die Rechenumgebung an ... 47
Schritt 5: Erstellen Sie eine Jobdefinition ... 48
Schritt 6: Reichen Sie einen Job ein ... 49
Schritt 7: Sehen Sie sich die Ausgabe des Jobs an ... 49
Schritt 8: (Optional) Reichen Sie einen Job mit Überschreibungen ein ... 50
Schritt 9: Bereinigen Sie Ihre Tutorial-Ressourcen ... 51
Weitere Ressourcen ... 52
Erste Schritte mit AWS Batch Amazon EKS Private Clusters ... 52
Übersicht ... 39
Voraussetzungen ... 55

Schritt 1: Erstellen Sie Ihren EKS-Cluster für AWS Batch ... 56
Schritt 2: Bereiten Sie Ihren EKS-Cluster vor für AWS Batch ... 58
Schritt 3: Erstellen Sie eine Amazon EKS-Rechenumgebung ... 61
Schritt 4: Erstellen Sie eine Job-Warteschlange und hängen Sie die Rechenumgebung an ... 63
Schritt 5: Erstellen Sie einen Amazon ECR mit Pull-Through-Cache ... 64
Schritt 6: Registrieren Sie eine Jobdefinition ... 65
Schritt 7: Reichen Sie einen Job zur Ausführung ein ... 66
Schritt 8: Sehen Sie sich die Ausgabe des Jobs an ... 66
Schritt 9: (Optional) Reichen Sie einen Job mit Überschreibungen ein ... 67
Schritt 10: Bereinigen Sie Ihre Tutorial-Ressourcen ... 68
Weitere Ressourcen ... 52
Fehlerbehebung ... 69
Das AWS Batch Widget-Dashboard ... 70
Fügen Sie das Widget „Einzelne Jobwarteschlange“ hinzu ... 70
Wie füge ich das CloudWatch Container Insights-Widget hinzu ... 71
Fügen Sie das Job-Logs-Widget hinzu ... 71
Rechenumgebungen für AWS Batch ... 73
Verwaltete Computerumgebungen ... 73
Überlegungen bei der Erstellung parallel Jobs mit mehreren Knoten ... 76
Nicht verwaltete Computerumgebungen ... 76
Erstellen Sie eine Rechenumgebung ... 77
Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mit Fargate-Ressourcen ... 78
Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von EC2 Amazon-Ressourcen ... 80
Tutorial: Erstellen Sie eine nicht verwaltete Rechenumgebung mithilfe von Amazon-Ressourcen EC2 ... 88
Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von Amazon EKS-Ressourcen ... 89
Ressource: Vorlage für die Rechenumgebung ... 94
Ressource berechnen AMIs ... 95
AMI-Spezifikation für Rechenressourcen ... 97
Tutorial: Erstellen Sie ein Compute-Ressourcen-AMI ... 99
Verwenden Sie ein GPU-Workload-AMI ... 102
Veraltete Amazon Linux-Version ... 108
Verwenden Sie EC2 Amazon-Startvorlagen ... 109
Startvorlagen als Standard festlegen und überschreiben ... 111

EC2 Amazon-Benutzerdaten in Startvorlagen ... 112
Referenz: Beispiele für Startvorlagen ... 113
EC2 Konfigurationen ... 116
Strategien für die Zuweisung von Instance-Typen ... 117
Speicherverwaltung ... 119
Systemspeicher reservieren ... 120
Tutorial: Rechenressourcenspeicher anzeigen ... 120
Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS ... 121
Aktualisierung von Rechenumgebungen ... 126
AWS Fargate Computerumgebungen aktualisieren ... 127
Aktualisierung der AMI-ID ... 130
Fargate-Computerumgebungen ... 131
Wann sollte Fargate verwendet werden ... 132
Jobdefinitionen bei Fargate ... 133
Warteschlangen für Job auf Fargate ... 135
Rechenumgebungen auf Fargate ... 135
Amazon EKS-Rechenumgebungen ... 136
Amazon EKS ... 139
Standard-AMI ... 140
Gemischte AMI-Umgebungen ... 140
Unterstützte Kubernetes-Versionen ... 141
Aktualisieren Sie die Kubernetes Version der Rechenumgebung ... 142
Gemeinsame Verantwortung der Kubernetes Knoten ... 143
Führen Sie eine DaemonSet auf AWS Batch verwalteten Knoten aus ... 144
Amazon EKS-Startvorlagen anpassen ... 144
Wie führe ich ein Upgrade von EKS AL2 auf EKS 023 AL2 durch ... 149
Warteschlangen für Job ... 151
Eine Job-Warteschlange erstellen ... 151
Tutorial: Erstellen Sie eine EC2 Amazon-Job-Warteschlange ... 152
Tutorial: Eine Fargate-Jobwarteschlange erstellen ... 153
Tutorial: Erstellen Sie eine Amazon EKS-Jobwarteschlange ... 154
Referenz: Vorlage für die Job-Warteschlange ... 155
Status der Auftragswarteschlange anzeigen ... 157
Informationen zur Job-Warteschlange anzeigen ... 157
Fair-Share-Richtlinien für die Terminplanung ... 158
Verwenden Sie Share-Identifikatoren ... 159

Verwenden Sie Planungsrichtlinien ... 160
Verwenden Sie Fair-Share-Scheduling ... 160
Tutorial: Eine Planungsrichtlinie erstellen ... 162
Referenz: Vorlage für Planungsrichtlinien ... 163
Ressourcenbewusste Planung ... 164
Schaffen Sie verbrauchbare Ressourcen ... 165
Geben Sie die Ressourcen für einen Job an ... 166
Überprüfen Sie die Ressourcennutzung ... 168
Aktualisieren Sie die Menge der verwendeten Ressourcen ... 170
Finden Sie Jobs, für die eine verbrauchbare Ressource erforderlich ist ... 171
Löschen Sie eine verbrauchbare Ressource ... 172
Berufsdefinitionen ... 174
Erstellen Sie eine Auftragsdefinition mit einem Knoten ... 174
Tutorial: Erstellen Sie eine Jobdefinition mit einem Knoten auf Amazon-Ressourcen EC2 ... 175
Tutorial: Erstellen Sie eine Einzelknoten-Jobdefinition auf Fargate-Ressourcen ... 182
Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten auf Amazon EKS-Ressourcen ... 188
Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten und mehreren Containern auf Amazon-Ressourcen EC2 ... 194
Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten ... 201
Tutorial: Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten auf Amazon-Ressourcen EC2 ... 202
Referenz: Jobdefinitionsvorlage mit ContainerProperties ... 210
Referenz: Jobdefinitionsparameter für ContainerProperties ... 217
Erstellen Sie Jobdefinitionen mit EcsProperties ... 264
`ContainerProperties`im Vergleich zu Jobdefinitionen `EcsProperties` ... 264
Allgemeine Änderungen an der AWS Batch APIs ... 265
Jobdefinitionen mit mehreren Containern für Amazon ECS ... 266
Jobdefinitionen mit mehreren Containern für Amazon EKS ... 267
Referenz: AWS Batch Jobszenarien mit EcsProperties ... 268
Verwenden Sie den awslogs-Protokolltreiber ... 274
awslogs-Protokolltreiberoptionen im Datentyp AWS Batch JobDefiniton ... 275
Geben Sie in Ihrer Jobdefinition eine Protokollkonfiguration an ... 278
Geben Sie sensible Daten an ... 279
Verwenden Sie Secrets Manager ... 280
Systems Manager Manager-Parameterspeicher verwenden ... 288

Authentifizierung in der privaten Registrierung für Jobs ... 293
Erforderliche IAM-Berechtigungen für die private Registrierungsauthentifizierung ... 294
Tutorial: Erstellen Sie ein Geheimnis für die Authentifizierung in privaten Registern ... 295
Amazon EFS-Volumes ... 296
Überlegungen zu Amazon EFS Volumes ... 297
Verwenden Sie Amazon EFS-Zugriffspunkte ... 298
Geben Sie in Ihrer Jobdefinition ein Amazon EFS-Dateisystem an ... 299
Referenz: Beispiele für Berufsdefinitionen ... 302
Referenz: Umgebungsvariablen ... 302
Referenz: Parametersetzung ... 303
Referenz: Testen Sie die GPU-Funktionalität ... 304
Referenz: Paralleljob mit mehreren Knoten ... 305
Aufträge ... 307
Tutorial: Einen Job einreichen ... 307
Auftragsstatus ... 311
Variablen der Arbeitsumgebung ... 314
Automatisierte Auftragswiederholungen ... 315
Abhängigkeiten von Job ... 317
Timeouts bei der Job ... 318
Stellenangebote bei Amazon EKS ... 319
Tutorial: Ordnen Sie einen laufenden Job einem Pod und einem Knoten zu ... 320
Tutorial: Ordnen Sie einen laufenden Pod seinem Job zu ... 321
parallel Jobs mit mehreren Knoten ... 323
Umgebungsvariablen ... 324
Knotengruppen ... 325
Lebenszyklus des Job ... 325
Überlegungen zur Rechenumgebung ... 326
parallel Jobs mit mehreren Knoten auf Amazon EKS ... 327
MNP-Jobs ausführen ... 328
Erstellen Sie eine Amazon EKS MNP-Jobdefinition ... 330
Einen Amazon EKS MNP-Job einreichen ... 332
Eine Amazon EKS MNP-Auftragsdefinition überschreiben ... 333
Ordnen Sie Jobs an ... 334
Referenz: Beispiel für einen Array-Job-Workflow ... 336
Praktische Einführung: Verwenden des Array-Job-Index ... 340
Führen Sie GPU-Jobs aus ... 346

Tutorial: Einen GPU-basierten Kubernetes Cluster auf Amazon EKS erstellen 351
Tutorial: Erstellen Sie eine Amazon EKS-GPU-Auftragsdefinition 353
Tutorial: Führen Sie einen GPU-Job in Ihrem Amazon EKS-Cluster aus 354
Suchen Sie nach AWS Batch Aufträgen in einer Jobwarteschlange 354
Tutorial: AWS Batch Jobs suchen und filtern 355
Tutorial: (Beta) AWS Batch Jobs suchen und filtern 356
Netzwerkmodi für AWS Batch Jobs 357
Tutorial: AWS Batch Jobprotokolle an CloudWatch Logs senden 359
Tutorial: AWS Batch Jobinformationen überprüfen 360
Sicherheit in AWS Batch 362
Identitäts- und Zugriffsverwaltung 363
Zielgruppe 363
Authentifizierung mit Identitäten 364
Verwalten des Zugriffs mit Richtlinien 368
Wie AWS Batch funktioniert mit IAM 371
Beispiele für identitätsbasierte Richtlinien 377
AWS verwaltete Richtlinien 380
IAM-Richtlinien, -Rollen und -Berechtigungen 396
Struktur der IAM-Richtlinien 396
Ressource: Beispielrichtlinien 400
Ressource: AWS Batch verwaltete Richtlinie 409
AWS Batch IAM-Ausführungsrolle 410
Unterstützte Berechtigungen auf Ressourcenebene 412
Tutorial: Erstellen Sie die IAM-Ausführungsrolle 412
Tutorial: Überprüfen Sie die IAM-Ausführungsrolle 413
Serviceverknüpfte Rollen verwenden 414
Amazon ECS-Instance-Rolle 422
Rolle bei Amazon EC2 Spot Fleet 426
EventBridge IAM-Rolle 429
Erstellen einer Virtual Private Cloud 431
Erstellen einer VPC 431
Nächste Schritte 432
VPC-Endpunkte 432
Überlegungen 433
Erstellen eines Schnittstellenendpunkts 434
Erstellen einer Endpunktrichtlinie 435

Compliance-Validierung ... 436
Sicherheit der Infrastruktur ... 437
Serviceübergreifende Confused-Deputy-Prävention ... 438
Beispiel: Rolle für den Zugriff auf nur eine Rechenumgebung ... 439
Beispiel: Rolle für den Zugriff auf mehrere Rechenumgebungen ... 440
CloudTrail ... 440
AWS Batch Informationen in CloudTrail ... 441
Referenz: Grundlegendes zu AWS Batch Protokolldateieinträgen ... 442
Problembehandlung bei IAM AWS Batch ... 444
Ich bin nicht autorisiert, eine Aktion in AWS Batch auszuführen. ... 444
Ich bin nicht berechtigt, iam auszuführen: PassRole ... 445
Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS Batch Ressourcen ermöglichen ... 445
AWS Step Functions ... 447
Tutorial: Details zur Zustandsmaschine anzeigen ... 447
Tutorial: Bearbeiten Sie eine Zustandsmaschine ... 448
Tutorial: Führen Sie eine Zustandsmaschine aus ... 448
Amazon EventBridge ... 449
AWS Batch Ereignisse ... 450
Ressource: Ereignisse zur Änderung des Berufsstatus ... 450
Ressource: Blockierte Ereignisse in der Jobwarteschlange ... 452
Tutorial: AWS Benutzerbenachrichtigungen verwenden mit AWS Batch ... 454
AWS Batch Jobs als EventBridge Ziele ... 454
Tutorial: Einen geplanten Job erstellen ... 456
Tutorial: Erstellen Sie eine Regel mit einem Ereignismuster ... 458
Tutorial: Eingangstransformator übergeben ... 461
Tutorial: Halten Sie Ausschau nach AWS Batch Job-Events ... 464
Voraussetzungen ... 464
Tutorial: Lambda-Funktion erstellen ... 465
Tutorial: Registrieren Sie die Ereignisregel ... 466
Tutorial: Testen Sie Ihre Konfiguration ... 468
Tutorial: Senden von Amazon Simple Notification Service-Benachrichtigungen für fehlgeschlagene Auftragsereignisse ... 468
Voraussetzungen ... 469
Tutorial: Ein Amazon SNS SNS-Thema erstellen und abonnieren ... 469
Tutorial: Registrieren Sie eine Ereignisregel ... 470

Tutorial: Testen Sie Ihre Regel ... 472
Alternative Regel: Batch-Job-Warteschlange blockiert ... 472
Elastic Fabric Adapter ... 474
Überwachen AWS Batch ... 477
CloudWatch Protokolle ... 478
Tutorial: Fügen Sie eine CloudWatch Logs-IAM-Richtlinie hinzu ... 479
Installieren und konfigurieren Sie den CloudWatch Agenten ... 480
Tutorial: CloudWatch Protokolle anzeigen ... 481
CloudWatch Einblicke in Container ... 483
Tutorial: Container Insights einschalten ... 483
Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen ... 484
Voraussetzungen ... 485
Installieren Sie das Add-on ... 485
Markieren Ihrer -Ressourcen mit Tags (Markierungen) ... 487
Grundlagen zu Tags (Markierungen) ... 487
Markieren Ihrer -Ressourcen mit Tags (Markierungen) ... 488
Tag-Einschränkungen ... 489
Tutorial: Tags mit der Konsole verwalten ... 490
Fügen Sie bei der Erstellung Tags zu einer einzelnen Ressource hinzu ... 490
Hinzufügen und Löschen von Tags (Markierungen) für einzelne Ressourcen ... 490
Tags mit der CLI oder API verwalten ... 491
Bewährte Methoden ... 493
Wann sollten Sie es verwenden AWS Batch ... 493
Checkliste für die skalierbare Ausführung ... 494
Optimieren Sie Container und AMIs ... 495
Wählen Sie die richtige Ressource für die Rechenumgebung ... 496
Amazon EC2 On-Demand oder Amazon EC2 Spot ... 497
Verwenden Sie die Best Practices von Amazon EC2 Spot für AWS Batch ... 498
Häufige Fehler und Problembehebung ... 500
Fehlerbehebung ... 504
AWS Batch ... 505
`INVALID`Datenverarbeitungsumgebung ... 505
Jobs, die in einem `RUNNABLE` Status feststecken ... 508
Spot-Instances wurden bei der Erstellung nicht markiert ... 514
Spot-Instances werden nicht herunterskaliert ... 514
Secrets Manager können nicht abgerufen werden ... 516

Die Ressourcenanforderungen für die Jobdefinition können nicht außer Kraft gesetzt werden 516
Fehlermeldung beim Aktualisieren der `desiredvCpus` Einstellung 518
AWS Batch auf Amazon EKS 518
`INVALID`Rechenumgebung 519
AWS Batch bei Amazon EKS bleibt der Job im `RUNNABLE` Status hängen 522
AWS Batch bei Amazon EKS bleibt der Job im `STARTING` Status hängen 523
Stellen Sie sicher, dass der richtig konfiguriert `aws-auth ConfigMap` ist 524
RBAC-Berechtigungen oder -Bindungen sind nicht richtig konfiguriert 525
Ressource: Servicekontingenten 528
Dokumentverlauf 530
..... dxxxvii

Was ist AWS Batch?

AWS Batch hilft Ihnen beim Ausführen von Batch-Computing-Workloads auf dem AWS Cloud. Batch-Computing ist eine gängige Methode für Entwickler, Wissenschaftler und Ingenieure, um auf große Mengen an Rechenressourcen zuzugreifen. AWS Batch macht die undifferenzierte, aufwändige Konfiguration und Verwaltung der erforderlichen Infrastruktur überflüssig, ähnlich wie bei herkömmlicher Batch-Computing-Software. Mit diesem Service können Sie Ressourcen als Reaktion auf gesendete Aufträge effizient bereitstellen, um Kapazitätsbeschränkungen aufzuheben, IT-Kosten zu reduzieren und Ergebnisse schnell bereitzustellen.

Dieser vollständig verwaltete Service AWS Batch unterstützt Sie bei der Ausführung von Batch-Computing-Workloads jeder Größenordnung. AWS Batch stellt Rechenressourcen automatisch bereit und optimiert die Verteilung der Arbeitslast auf der Grundlage der Menge und des Umfangs der Workloads. Mit AWS Batch müssen Sie keine Batch-Computing-Software installieren oder verwalten, sodass Sie sich voll und ganz auf die Analyse der Ergebnisse und die Lösung von Problemen konzentrieren können.

AWS Batch bietet alle erforderlichen Funktionen, um umfangreiche, rechenintensive Workloads auf Basis von AWS Managed Container Orchestration Services, Amazon ECS und Amazon EKS auszuführen. AWS Batch ist in der Lage, die Rechenkapazität auf EC2 Amazon-Instances und Fargate-Ressourcen zu skalieren.

AWS Batch bietet einen vollständig verwalteten Service für Batch-Workloads und stellt die Betriebsfunktionen bereit, um diese Arten von Workloads im Hinblick auf Durchsatz, Geschwindigkeit, Ressourceneffizienz und Kosten zu optimieren.

Themen

- Sind Sie ein AWS Batch Erstanwender?
- Zugehörige Services
- Zugreifen AWS Batch
- Bestandteile von AWS Batch

Sind Sie ein AWS Batch Erstanwender?

Wenn Sie zum ersten Mal Benutzer von sind AWS Batch, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- Bestandteile von AWS Batch
- IAM-Konto und Administratorbenutzer erstellen
- Einrichten AWS Batch
- Erste Schritte mit AWS Batch Tutorials

Zugehörige Services

AWS Batch ist ein vollständig verwalteter Batch-Computing-Service, der Ihre containerisierten Batch-ML-, Simulations- und Analyse-Workloads für die gesamte Palette von AWS Rechenangeboten wie Amazon ECS, Amazon EKS und Spot- oder On-Demand-Instances plant AWS Fargate, terminiert und ausführt. Weitere Informationen zu den einzelnen verwalteten Rechendiensten finden Sie unter:

- EC2 Amazon-Benutzerhandbuch
- AWS Fargate-Entwicklerhandbuch
- Amazon EKS-Benutzerhandbuch

Zugreifen AWS Batch

Sie können wie folgt darauf AWS Batch zugreifen:

AWS Batch console

Die Weboberfläche, auf der Sie Ressourcen erstellen und verwalten.

AWS Command Line Interface

Interagieren AWS-Services Sie mithilfe von Befehlen in Ihrer Befehlszeilen-Shell. Das AWS Command Line Interface wird unter Windows, MacOS und Linux unterstützt. Weitere Informationen zu finden Sie im AWS CLIAWS Command Line Interface Benutzerhandbuch. Sie finden die AWS Batch Befehle in der AWS CLI Befehlsreferenz.

AWS SDKs

Wenn Sie es vorziehen, Anwendungen sprachspezifisch zu erstellen, APIs anstatt eine Anfrage über HTTP oder HTTPS einzureichen, verwenden Sie die Bibliotheken, den Beispielcode, die Tutorials und andere Ressourcen von. AWS Diese Bibliotheken bieten grundlegende Funktionen zur Automatisierung von Aufgaben, wie z. B. das kryptografische Signieren Ihrer Anfragen, das Wiederholen von Anfragen und das Behandeln von Fehlerantworten. Diese Funktionen machen den Einstieg für Sie effizienter. Weitere Informationen finden Sie unter Tools, auf denen Sie aufbauen können AWS.

Bestandteile von AWS Batch

AWS Batch vereinfacht die Ausführung von Batch-Jobs in mehreren Availability Zones innerhalb einer Region. Sie können AWS Batch -Datenverarbeitungsumgebungen in einer neuen oder bestehenden VPC erstellen. Nachdem eine Datenverarbeitungsumgebung erstellt und einer Auftragswarteschlange zugewiesen wurde, können Sie Auftragsdefinitionen definieren, die Docker-Containerabbilder für die Ausführung Ihrer Aufträge festzulegen. Containerabbilder werden in Container-Registrys gespeichert und daraus abgerufen, die sich innerhalb oder außerhalb Ihrer AWS -Infrastruktur befinden können.

Datenverarbeitungsumgebung

Bei einer Datenverarbeitungsumgebung handelt es sich um eine Gruppe verwalteter oder nicht verwalteter Datenverarbeitungsressourcen für die Ausführung von Aufträgen. In verwalteten Rechenumgebungen können Sie den gewünschten Berechnungstyp (Fargate oder EC2) auf mehreren Detailebenen angeben. Sie können Rechenumgebungen einrichten, die einen bestimmten EC2 Instanztyp, ein bestimmtes Modell wie `c5.2xlarge` oder `m5.10xlarge` verwenden. Oder Sie können nur angeben, dass Sie die neuesten Instanztypen verwenden möchten. Sie können auch die minimale, gewünschte und maximale Anzahl von v CPUs für die Umgebung sowie den Betrag angeben, den Sie bereit sind, für eine Spot-Instance als Prozentsatz des On-Demand-Instance-Preises zu zahlen, und einen Zielsatz von VPC-Subnetzen. AWS Batch Startet,

verwaltet und beendet Compute-Typen effizient nach Bedarf. Sie können zudem Ihre eigenen Datenverarbeitungsumgebungen verwalten. Daher sind Sie für die Einrichtung und Skalierung der Instances in einem Amazon ECS-Cluster verantwortlich, der für Sie AWS Batch erstellt. Weitere Informationen finden Sie unter Rechenumgebungen für AWS Batch.

Warteschlangen für Job

Wenn Sie einen AWS Batch Job einreichen, senden Sie ihn an eine bestimmte Job-Warteschlange, in der der Job gespeichert wird, bis er in einer Computerumgebung eingeplant wird. Sie ordnen einer Auftragswarteschlange eine oder mehrere Rechenumgebungen zu. Sie können diesen Rechenumgebungen und sogar allen Job-Warteschlangen selbst Prioritätswerte zuweisen. Sie können beispielsweise eine Warteschlange mit hoher Priorität einrichten, an die Sie zeitkritische Jobs senden, und eine Warteschlange mit niedriger Priorität für Jobs, die jederzeit ausgeführt werden können, wenn Rechenressourcen günstiger sind. Weitere Informationen finden Sie unter Warteschlangen für Job.

Berufsdefinitionen

Eine Jobdefinition gibt an, wie Jobs ausgeführt werden sollen. Sie können sich eine Jobdefinition als Blaupause für die Ressourcen in Ihrem Job vorstellen. Sie können Ihrem Job eine IAM-Rolle zuweisen, um Zugriff auf andere AWS Ressourcen zu gewähren. Sie geben auch sowohl Speicher- als auch CPU-Anforderungen an. Mit der Auftragsdefinition können zudem Containereigenschaften, Umgebungsvariablen und Bereitstellungspunkte zur persistente Speicherung gesteuert werden. Viele der Spezifikationen in einer Auftragsdefinition können überschrieben werden, indem Sie beim Senden einzelner Aufträge neue Werte angeben. Weitere Informationen finden Sie unter Berufsdefinitionen.

Aufträge

Eine Arbeitseinheit (etwa als Shell-Script, eine ausführbare Linux-Datei oder ein Docker-Containerabbild), die Sie an AWS Batch senden. Es hat einen Namen und wird als containerisierte Anwendung auf AWS Fargate oder EC2 Amazon-Ressourcen in Ihrer Rechenumgebung ausgeführt, wobei Parameter verwendet werden, die Sie in einer Jobdefinition angeben. Jobs können mit Namen oder ID auf andere Jobs verweisen und können vom erfolgreichen Abschluss anderer Jobs oder der Verfügbarkeit der von Ihnen angegebenen Ressourcen abhängig sein. Weitere Informationen finden Sie unter Aufträge.

Richtlinie zur Terminplanung

Mithilfe von Planungsrichtlinien können Sie konfigurieren, wie Rechenressourcen in einer Auftragswarteschlange Benutzern oder Workloads zugewiesen werden. Mithilfe von Fair-Share-Scheduling-Richtlinien können Sie Workloads oder Benutzern unterschiedliche Share-IDs zuweisen. Der AWS Batch Job Scheduler verwendet standardmäßig eine FIFO-Strategie (First-In, First Out). Weitere Informationen finden Sie unter Fair-Share-Terminplanungsrichtlinien.

Verbrauchbare Ressourcen

Eine verbrauchbare Ressource ist eine Ressource, die für die Ausführung Ihrer Jobs benötigt wird, z. B. ein Lizenz-Token eines Drittanbieters, die Bandbreite für den Datenbankzugriff, die Notwendigkeit, Aufrufe einer Drittanbieter-API zu drosseln usw. Sie geben die verbrauchbaren Ressourcen an, die für die Ausführung eines Jobs benötigt werden, und Batch berücksichtigt diese Ressourcenabhängigkeiten bei der Planung eines Jobs. Sie können die Unterauslastung der Rechenressourcen verringern, indem Sie nur die Jobs zuweisen, für die alle erforderlichen Ressourcen verfügbar sind. Weitere Informationen finden Sie unter Ressourcenbewusste Planung .

Einrichten AWS Batch

Wenn Sie sich bereits für Amazon Web Services (AWS) registriert haben und Amazon Elastic Compute Cloud (Amazon EC2) oder Amazon Elastic Container Service (Amazon ECS) verwenden, können Sie dies bald nutzen AWS Batch. Der Einrichtungsprozess für diese Dienste ist ähnlich. Dies liegt daran, dass Amazon ECS-Container-Instances in seinen Rechenumgebungen AWS Batch verwendet werden. Um with verwenden zu können AWS Batch, müssen Sie eine Version von verwenden AWS CLI , die die neuesten AWS Batch Funktionen unterstützt. AWS CLI Wenn Sie in der keine Unterstützung für eine AWS Batch Funktion sehen AWS CLI, führen Sie ein Upgrade auf die neueste Version durch. Weitere Informationen finden Sie unter http://aws.amazon.com/cli/.

> **Note**
>
> Da Komponenten von Amazon AWS Batch verwendet werden EC2, verwenden Sie für viele dieser Schritte die EC2 Amazon-Konsole.

Erledigen Sie die folgenden Aufgaben, um sich darauf vorzubereiten AWS Batch.

Themen

- IAM-Konto und Administratorbenutzer erstellen
- Erstellen Sie IAM-Rollen für Ihre Rechenumgebungen und Container-Instances
- Erstellen Sie ein key pair für Ihre Instances
- Erstellen einer VPC
- Eine Sicherheitsgruppe erstellen
- Installiere das AWS CLI

IAM-Konto und Administratorbenutzer erstellen

Um zu beginnen, müssen Sie ein AWS Konto und einen einzelnen Benutzer erstellen, dem in der Regel Administratorrechte gewährt werden. Um dies zu erreichen, führen Sie die folgenden Tutorials durch:

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Ein Teil des Anmeldevorgangs umfasst den Empfang eines Telefonanrufs oder einer Textnachricht und die Eingabe eines Bestätigungscodes auf der Telefontastatur.

   Wenn Sie sich für eine anmelden AWS-Konto, wird eine Root-Benutzer des AWS-Kontoserstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

   Anweisungen finden Sie unter Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer (Konsole) im IAM-Benutzerhandbuch.

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie IAM-Identity-Center-Verzeichnis im Benutzerhandbuch unter Benutzerzugriff mit der Standardeinstellung konfigurieren.AWS IAM Identity Center

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Erstellen Sie IAM-Rollen für Ihre Rechenumgebungen und Container-Instances

Ihre AWS Batch Rechenumgebungen und Container-Instances benötigen AWS-Konto Anmeldeinformationen, um in Ihrem Namen Anrufe AWS APIs an andere zu tätigen. Erstellen Sie eine AWS Identity and Access Management Rolle, die diese Anmeldeinformationen für Ihre Rechenumgebungen und Container-Instances bereitstellt, und ordnen Sie diese Rolle dann Ihren Rechenumgebungen zu.

> **Note**
>
> Informationen dazu, ob Sie AWS-Konto über die erforderlichen Berechtigungen verfügen, finden Sie unter Erste Einrichtung des IAM-Dienstes für Ihr Konto.
> Die Rollen AWS Batch Rechenumgebung und Container-Instance werden bei der ersten Ausführung der Konsole automatisch für Sie erstellt. Wenn Sie also beabsichtigen, die AWS Batch Konsole zu verwenden, können Sie mit dem nächsten Abschnitt fortfahren. Wenn Sie AWS CLI stattdessen die verwenden möchten, führen Sie die Verfahren unterVerwenden Sie dienstbezogene Rollen für AWS Batch, und ausAmazon ECS-Instance-Rolle, Tutorial: Erstellen Sie die IAM-Ausführungsrolle bevor Sie Ihre erste Computerumgebung erstellen.

Erstellen Sie ein key pair für Ihre Instances

AWS verwendet Public-Key-Kryptografie, um die Anmeldeinformationen für Ihre Instance zu sichern. Eine Linux-Instance, z. B. eine AWS Batch Compute-Environment-Container-Instance, hat kein Passwort, das für den SSH-Zugriff verwendet werden könnte. Sie verwenden ein Schlüsselpaar, um sich sicher an Ihrer Instance anzumelden. Sie geben beim Erstellen Ihrer Datenverarbeitungsumgebung den Namen des Schlüsselpaares an und stellen dann beim Anmelden über SSH den privaten Schlüssel bereit.

Wenn Sie noch kein key pair erstellt haben, können Sie eines mit der EC2 Amazon-Konsole erstellen. Beachten Sie, dass Sie, wenn Sie planen, Instances in mehreren zu starten AWS-Regionen, in jeder Region ein key pair erstellen müssen. Weitere Informationen zu Regionen finden Sie unter Regionen und Availability Zones im EC2 Amazon-Benutzerhandbuch.

So erstellen Sie ein Schlüsselpaar

1. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.

2. Wählen Sie in der Navigationsleiste eine AWS-Region für das key pair aus. Sie können unabhängig von Ihrem Standort jede Region auswählen, die Ihnen zur Verfügung steht. Schlüsselpaare sind jedoch regionsspezifisch. Wenn Sie beispielsweise planen, eine Instance in der Region USA West (Oregon) zu starten, erstellen Sie ein key pair für die Instance in derselben Region.
3. Wählen Sie im Navigationsbereich Key Pairs (Schlüsselpaare) und Create Key Pair (Schlüsselpaar erstellen) aus.
4. Geben Sie im Dialogfeld Create Key Pair (Schlüsselpaar erstellen) im Feld Key pair name (Name des Schlüsselpaars) einen Namen für das neue Schlüsselpaar ein und klicken Sie auf Erstellen. Wählen Sie einen Namen, an den Sie sich erinnern können, z. B. Ihren Benutzernamen, gefolgt von `-key-pair`, plus dem Namen der Region. Beispiel: me-key-pair-uswest2.
5. Die private Schlüsseldatei wird von Ihrem Browser automatisch runtergeladen. Der Basisdateiname ist der Name, den Sie als Namen für Ihr Schlüsselpaar festgelegt haben, und die Dateinamenerweiterung lautet `.pem`. Speichern Sie die Datei mit dem privaten Schlüssel an einem sicheren Ort.

   > **Important**
   >
   > Dies ist die einzige Möglichkeit, die private Schlüsseldatei zu speichern. Sie müssen den Namen Ihres key pair angeben, wenn Sie eine Instance starten, und den entsprechenden privaten Schlüssel jedes Mal, wenn Sie sich mit der Instance verbinden.

6. Wenn Sie einen SSH-Client auf einem Mac- oder Linux-Computer verwenden, um eine Verbindung zu Ihrer Linux-Instance herzustellen, verwenden Sie den folgenden Befehl, um die Berechtigungen für Ihre private Schlüsseldatei festzulegen. Auf diese Weise können nur Sie es lesen.

   ```
   $ chmod 400 your_user_name-key-pair-region_name.pem
   ```

Weitere Informationen finden Sie unter Amazon EC2 Key Pairs im EC2 Amazon-Benutzerhandbuch.

So stellen Sie über Ihr Schlüsselpaar eine Verbindung zu Ihrer Instance her

Wenn Sie von einem Computer mit Mac oder Linux eine Verbindung zu Ihrer Linux-Instance herstellen möchten, geben Sie die `.pem`-Datei für Ihren SSH-Client mit der Option `-i` und dem Pfad zu Ihrem privaten Schlüssel an. Um von einem Computer unter Windows aus eine Verbindung zu Ihrer Linux-Instance herzustellen, verwenden Sie entweder MindTerm oder PuTTY. Wenn Sie PuTTY

verwenden möchten, installieren Sie es und konvertieren Sie die `.pem` Datei wie folgt in eine `.ppk` Datei.

(Optional) So bereiten Sie die Verbindung zu einer Linux-Instance von Windows aus mithilfe von PuTTY vor

1. Laden Sie PuTTY unter http://www.chiark.greenend.org.uk/~sgtatham/putty/ herunter und installieren Sie die Anwendung. Vergewissern Sie sich, dass Sie die gesamte Suite installieren.
2. Starten Sie Pu TTYgen (wählen Sie beispielsweise im Startmenü Alle Programme, PuTTY und Pu TTYgen).
3. Wählen Sie unter Type of key to generate die Option RSA. Wenn Sie eine frühere Version von Pu verwendenTTYgen, wählen Sie SSH-2 RSA.

4. Wählen Sie Laden aus. Standardmäßig TTYgen zeigt Pu nur Dateien mit der Erweiterung `.ppk` an. Damit Sie die `.pem`-Datei finden, wählen Sie die Option zur Anzeige aller Dateitypen aus.

5. Wählen Sie die Datei mit dem privaten Schlüssel aus, die Sie im vorherigen Schritt erstellt haben, und klicken Sie auf Open. Klicken Sie auf OK, um das Bestätigungsdialogfeld zu verlassen.
6. Wählen Sie Save private key (Privaten Schlüssel speichern) aus. Pu TTYgen zeigt eine Warnung zum Speichern des Schlüssels ohne Passphrase an. Wählen Sie Yes (Ja).
7. Geben Sie für den Schlüssel denselben Namen an wie für das Schlüsselpaar. PuTTY fügt automatisch die Dateierweiterung `.ppk` hinzu.

Erstellen einer VPC

Mit Amazon Virtual Private Cloud (Amazon VPC) können Sie AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk starten. Wir empfehlen dringend, dass Sie Ihre Container-Instances in einer VPC starten.

Wenn Sie eine Standard-VPC haben, können Sie diesen Abschnitt auch überspringen und mit der nächsten Aufgabe Eine Sicherheitsgruppe erstellen fortfahren. Informationen darüber, ob Sie über eine Standard-VPC verfügen, finden Sie im EC2 Amazon-Benutzerhandbuch unter Unterstützte Plattformen in der EC2 Amazon-Konsole.

Informationen zum Erstellen einer Amazon VPC finden Sie unter Nur VPC erstellen im Amazon VPC-Benutzerhandbuch. In der folgenden Tabelle können Sie ermitteln, welche Optionen Sie auswählen müssen.

Option	Wert
Zu erstellende Ressourcen	Nur VPC
Name	Geben Sie optional einen Namen für Ihre VPC ein.
IPv4 CIDR-Block	IPv4 Manuelle CIDR-Eingabe Die CIDR-Blockgröße muss eine Größe zwischen /16 und /28. haben.
IPv6 CIDR-Block	Kein IPv6 CIDR-Block
Tenancy	Standard

Weitere Informationen über Amazon VPC finden Sie unter Was ist Amazon VPC? im Amazon VPC-Benutzerhandbuch.

Eine Sicherheitsgruppe erstellen

Sicherheitsgruppen fungieren als Firewall für zugehörige Container-Instances in der Datenverarbeitungsumgebung. Sie kontrollieren den ein- und ausgehenden Datenverkehr auf Container-Instance-Ebene. Eine Sicherheitsgruppe kann nur in der VPC verwendet werden, für die sie erstellt wird.

Sie können einer Sicherheitsgruppe Regeln hinzufügen, die es Ihnen ermöglichen, von Ihrer IP-Adresse über SSH eine Verbindung zu Ihrer Container-Instance herzustellen. Sie können auch

Regeln hinzufügen, die den ein- und ausgehenden HTTP- und HTTPS-Zugriff von überall zulassen. Fügen Sie alle Regeln zum Öffnen von Ports hinzu, die für Ihre Aufgaben benötigt werden.

Beachten Sie, dass Sie, wenn Sie Container-Instances in mehreren Regionen starten möchten, in jeder Region eine Sicherheitsgruppe erstellen müssen. Weitere Informationen finden Sie unter Regionen und Availability Zones im EC2 Amazon-Benutzerhandbuch.

> **Note**
>
> Sie benötigen die öffentliche IP-Adresse Ihres lokalen Computers, die Sie mithilfe eines Service abrufen können. Wir bieten beispielsweise den folgenden Service an: http://checkip.amazonaws.com/ oder https://checkip.amazonaws.com/. Wenn Sie einen anderen Service suchen möchten, der Ihnen Ihre IP-Adresse nennt, verwenden Sie den Suchausdruck "wie ist meine IP-Adresse". Wenn Sie die Verbindung über einen Internetdienstanbieter (ISP) oder hinter einer Firewall ohne statische IP-Adresse herstellen, ermitteln Sie den Bereich der IP-Adressen, die von Client-Computern verwendet werden.

Erstellen einer Sicherheitsgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon-VPC-Konsole unter https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
3. Wählen Sie Create security group (Sicherheitsgruppe erstellen) aus.
4. Geben Sie einen Namen und eine Beschreibung für die Sicherheitsgruppe ein. Sie können den Namen und die Beschreibung einer Regelgruppe nach der Erstellung nicht mehr ändern.
5. Wählen Sie unter VPC die VPC aus.
6. (Optional) Standardmäßig beginnen neue Sicherheitsgruppen nur mit einer Regel für ausgehenden Datenverkehr, die es erlaubt, dass der gesamte Datenverkehr die Ressource verlässt. Sie müssen Regeln hinzufügen, um eingehenden Datenverkehr zuzulassen oder den ausgehenden Datenverkehr einzuschränken.

   AWS Batch Für Container-Instances müssen keine eingehenden Ports geöffnet sein. Möglicherweise möchten Sie jedoch eine SSH-Regel hinzufügen. Auf diese Weise können Sie sich bei der Container-Instance anmelden und die Container in Jobs mit Docker-Befehlen untersuchen. Wenn Sie möchten, dass Ihre Container-Instance einen Job hostet, der einen Webserver ausführt, können Sie auch Regeln für HTTP hinzufügen. Führen Sie die folgenden Schritte aus, um diese optionale Regeln für die Sicherheitsgruppe hinzuzufügen.

Erstellen Sie auf der Registerkarte Eingehend die folgenden Regeln und wählen Sie Erstellen aus:

- Klicken Sie auf Add Rule (Regel hinzufügen). Wählen Sie für Typ die Option HTTP aus. Wählen Sie unter Quelle die Option Anywhere (Beliebig) (`0.0.0.0/0`) aus.
- Klicken Sie auf Add Rule (Regel hinzufügen). Wählen Sie unter Typ die Option SSH aus. Wählen Sie für Source die Option Custom IP und geben Sie die öffentliche IP-Adresse Ihres Computers oder Netzwerks in der CIDR-Notation (Classless Inter-Domain Routing) an. Wenn Ihr Unternehmen Adressen aus einem Bereich zuweist, geben Sie den gesamten Bereich an, z. B. `203.0.113.0/24`. Um eine individuelle IP-Adresse in CIDR-Notation anzugeben, wählen Sie Meine IP aus. Dadurch wird das Routing-Präfix `/32` zur öffentlichen IP-Adresse hinzugefügt.

> **Note**
>
> Aus Sicherheitsgründen empfehlen wir nicht, SSH-Zugriff von allen IP-Adressen (`0.0.0.0/0`) auf Ihre Instance zuzulassen, sondern nur zu Testzwecken und nur für kurze Zeit.

7. Sie können Tags (Markierungen) jetzt erstellen oder zu einem späteren Zeitpunkt hinzufügen. Um eine Markierung hinzuzufügen, wählen Sie Add Tags (Tags (Markierung) hinzufügen) und geben Sie dann den Markierungsschlüssel und -Wert ein.
8. Wählen Sie Sicherheitsgruppe erstellen aus.

Informationen zum Erstellen einer Sicherheitsgruppe über die Befehlszeile finden Sie unter create-security-group()AWS CLI

Weitere Informationen zu Sicherheitsgruppen finden Sie unter Arbeiten mit Sicherheitsgruppen.

Installiere das AWS CLI

Um das AWS CLI mit zu verwenden AWS Batch, installieren Sie die neueste AWS CLI Version. Informationen zur Installation AWS CLI oder zum Upgrade auf die neueste Version finden Sie unter Installation der AWS Befehlszeilenschnittstelle im AWS Command Line Interface Benutzerhandbuch.

Erste Schritte mit AWS Batch Tutorials

Sie können den Assistenten für die AWS Batch erste Ausführung verwenden, um schnell loszulegen. AWS Batch Nachdem Sie die Voraussetzungen erfüllt haben, können Sie den Assistenten für die erste Ausführung verwenden, um eine Rechenumgebung, eine Auftragsdefinition und eine Auftragswarteschlange zu erstellen.

Sie können auch einen Beispieljob für „Hello World“ einreichen, indem Sie den Assistenten für die AWS Batch erste Ausführung verwenden, um Ihre Konfiguration zu testen. Wenn Sie bereits über ein Docker-Image verfügen, in dem Sie starten möchten AWS Batch, können Sie dieses Image verwenden, um eine Jobdefinition zu erstellen.

Anschließend können Sie mit dem Assistenten für die AWS Batch erste Ausführung eine Rechenumgebung und eine Job-Warteschlange erstellen und einen Hello World-Beispieljob einreichen.

Erste Schritte mit Amazon EC2 Orchestration mithilfe des Wizards

Amazon Elastic Compute Cloud (Amazon EC2) bietet skalierbare Rechenkapazität in der AWS Cloud. EC2Durch die Nutzung von Amazon müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können.

Sie können Amazon verwenden EC2 , um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten. Amazon EC2 ermöglicht es Ihnen, nach oben oder unten zu skalieren, um Änderungen der Anforderungen oder Beliebtheitsspitzen zu bewältigen, sodass Sie weniger Traffic prognostizieren müssen.

Übersicht

Dieses Tutorial zeigt, wie Sie Amazon AWS Batch mit dem Wizard einrichten EC2 und ausführen`Hello World`.

Beabsichtigte Zielgruppe

Dieses Tutorial richtet sich an Systemadministratoren und Entwickler, die für die Einrichtung, das Testen und die Bereitstellung verantwortlich sind AWS Batch.

Verwendete Funktionen

In diesem Tutorial erfahren Sie, wie Sie den AWS Batch Konsolenassistenten verwenden, um:

- Erstellen und konfigurieren Sie eine EC2 Amazon-Rechenumgebung
- Erstellen Sie eine Job-Warteschlange.
- Erstellen Sie eine Auftragsdefinition
- Erstellen Sie einen Job und reichen Sie ihn zur Ausführung ein
- Sehen Sie sich die Ausgabe des Jobs an in CloudWatch

Erforderliche Zeit

Die Fertigstellung dieses Tutorials sollte etwa 10—15 Minuten dauern.

Regionale Einschränkungen

Es gibt keine landesspezifischen oder regionalen Einschränkungen im Zusammenhang mit der Verwendung dieser Lösung.

Kosten für die Ressourcennutzung

Für die Erstellung eines AWS Kontos fallen keine Gebühren an. Durch die Implementierung dieser Lösung können Ihnen jedoch einige oder alle Kosten entstehen, die in der folgenden Tabelle aufgeführt sind.

Beschreibung	Kosten (US-Dollar)
EC2 Amazon-Instanz	Sie zahlen für jede EC2 Amazon-Instance, die erstellt wird. Weitere Informationen zur Preisgestaltung finden Sie unter EC2 Amazon-Preise.

Voraussetzungen

Bevor Sie beginnen:

- Erstellen Sie eine AWS-Konto , falls Sie noch keine haben.
- Erstellen Sie die `ecsInstanceRole`Instanzrolle.

Schritt 1: Erstellen Sie eine Rechenumgebung

> **Important**
>
> Um so einfach und schnell wie möglich loszulegen, enthält dieses Tutorial Schritte mit Standardeinstellungen. Wir empfehlen Ihnen, sich vor der Erstellung für den produktiven Einsatz mit allen Einstellungen vertraut zu machen und die Bereitstellung mit den Einstellungen vorzunehmen, die Ihren Anforderungen entsprechen.

Gehen Sie wie folgt vor, um eine Rechenumgebung für eine EC2 Amazon-Orchestrierung zu erstellen:

1. Öffnen Sie den Assistenten für die erste Ausführung der AWS Batch Konsole.
2. Wählen Sie für Job und Orchestrierungstyp konfigurieren die Option Amazon Elastic Compute Cloud (Amazon EC2).
3. Wählen Sie Weiter aus.
4. Geben Sie im Abschnitt Konfiguration der Rechenumgebung für Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
5. Wählen Sie unter Instanzrolle eine bestehende Instanzrolle aus, der die erforderlichen IAM-Berechtigungen zugewiesen sind. Diese Instance-Rolle ermöglicht es den Amazon ECS-Container-Instances in Ihrer Rechenumgebung, die erforderlichen AWS API-Operationen aufzurufen. Weitere Informationen finden Sie unter Amazon ECS-Instance-Rolle.

   Der Standardname der Instance-Rolle lautet`ecsInstanceRole`.
6. Für die Instanzkonfiguration können Sie die Standardeinstellungen beibehalten.
7. Verwenden Sie für die Netzwerkkonfiguration Ihre Standard-VPC für. AWS-Region
8. Wählen Sie Weiter aus.

Schritt 2: Erstellen Sie eine Job-Warteschlange

In einer Job-Warteschlange werden Ihre eingereichten Jobs gespeichert, bis der AWS Batch Scheduler den Job auf einer Ressource in Ihrer Computerumgebung ausführt. Weitere Informationen finden Sie unter Warteschlangen für Job.

Gehen Sie wie folgt vor, um eine Job-Warteschlange für eine EC2 Amazon-Orchestrierung zu erstellen:

1. Geben Sie Job Job-Queue-Konfiguration für Name einen eindeutigen Namen für Ihre Job-Warteschlange an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
2. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
3. Wählen Sie Weiter aus.

Schritt 3: Erstellen Sie eine Jobdefinition

AWS Batch Jobdefinitionen geben an, wie Jobs ausgeführt werden sollen. Obwohl jeder Job auf eine Auftragsdefinition verweisen muss, können viele der in der Auftragsdefinition angegebenen Parameter zur Laufzeit außer Kraft gesetzt werden.

So erstellen Sie die Jobdefinition:

1. Für Erstellen Sie eine Jobdefinition
   a. Geben Sie für Name einen eindeutigen Namen für Ihre Job-Warteschlange ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
   b. Unter Befehl — optional können Sie `hello world` zu einer benutzerdefinierten Nachricht wechseln oder sie unverändert lassen.
2. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
3. Wählen Sie Weiter aus.

Schritt 4: Erstellen Sie einen Job

Gehen Sie wie folgt vor, um einen Job zu erstellen:

1. Geben Sie im Abschnitt Auftragskonfiguration für Name einen eindeutigen Namen für den Job an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
2. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
3. Wählen Sie Weiter aus.

Schritt 5: Überprüfen und Erstellen

Überprüfen Sie auf der Seite Überprüfen und erstellen die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Ressourcen erstellen aus.

1. Wählen Sie unter Überprüfen und Erstellen die Option Ressourcen erstellen aus.
2. Es öffnet sich ein Fenster, in dem Sie AWS Batch beginnen, Ihre Ressourcen zuzuweisen. Sobald Sie fertig sind, wählen Sie Gehe zum Dashboard. Auf dem Dashboard sollten Sie alle Ihnen zugewiesenen Ressourcen sehen und sehen, dass sich der Job im `Runnable` Status befindet. Ihr Job ist für die Ausführung geplant und sollte in 2—3 Minuten abgeschlossen sein.

Schritt 6: Sehen Sie sich die Ausgabe des Jobs an

Gehen Sie wie folgt vor, um die Ausgabe des Jobs anzuzeigen:

1. Wählen Sie im Navigationsbereich Jobs aus.
2. Wählen Sie im Drop-down-Menü Job-Warteschlange die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.
3. In der Tabelle Jobs werden all Ihre Jobs und ihr aktueller Status aufgeführt. Sobald der Status des Job erfolgreich ist, wählen Sie den Namen des Jobs, um die Details des Jobs anzuzeigen.
4. Wählen Sie im Detailbereich den Namen des Protokollstreams aus. Die CloudWatch Konsole für den Job wird geöffnet und es sollte ein Ereignis mit der Nachricht von `hello world` oder Ihrer benutzerdefinierten Nachricht geben.

Schritt 7: Bereinigen Sie Ihre Tutorial-Ressourcen

Die EC2 Amazon-Instance wird Ihnen in Rechnung gestellt, solange sie aktiviert ist. Sie können die Instance löschen, damit keine Gebühren mehr anfallen.

Gehen Sie wie folgt vor, um die von Ihnen erstellten Ressourcen zu löschen:

1. Wählen Sie im Navigationsbereich die Option Job-Warteschlange aus.
2. Wählen Sie in der Tabelle Job-Queue die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.

3. Wählen Sie Disable (deaktivieren) aus. Sobald der Status der Job-Warteschlange deaktiviert ist, können Sie Löschen wählen.
4. Sobald die Job-Warteschlange gelöscht ist, wählen Sie im Navigationsbereich Computing-Umgebungen aus.
5. Wählen Sie die Rechenumgebung aus, die Sie für dieses Tutorial erstellt haben, und klicken Sie dann auf Deaktivieren. Es kann 1—2 Minuten dauern, bis die Deaktivierung der Rechenumgebung abgeschlossen ist.
6. Sobald der Status der Rechenumgebung Deaktiviert ist, wählen Sie Löschen aus. Es kann 1—2 Minuten dauern, bis die Rechenumgebung gelöscht ist.

Weitere Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, möchten Sie sich möglicherweise mit den folgenden Themen befassen:

- Erkunden Sie die AWS Batch Kernkomponenten. Weitere Informationen finden Sie unter Bestandteile von AWS Batch.
- Erfahren Sie mehr über die verschiedenen verfügbaren Rechenumgebungen unter AWS Batch.
- Erfahren Sie mehr über Job-Warteschlangen und ihre verschiedenen Planungsoptionen.
- Erfahren Sie mehr über Jobdefinitionen und die verschiedenen Konfigurationsoptionen.
- Erfahren Sie mehr über die verschiedenen Arten von Jobs.

Erste Schritte mit AWS Batch und Fargate-Orchestrierung mithilfe des Wizards

AWS Fargate startet und skaliert die Berechnung so, dass sie den Ressourcenanforderungen, die Sie für den Container angeben, genau entspricht. Mit Fargate müssen Sie nicht zu viele Server bereitstellen oder für zusätzliche Server bezahlen. Weitere Informationen finden Sie unter Fargate.

Übersicht

Dieses Tutorial zeigt, wie Sie AWS Fargate AWS Batch mit dem Wizard einrichten und ausführen`Hello World`.

Beabsichtigte Zielgruppe

Dieses Tutorial richtet sich an Systemadministratoren und Entwickler, die für die Einrichtung, das Testen und die Bereitstellung verantwortlich sind AWS Batch.

Verwendete Funktionen

In diesem Tutorial erfahren Sie, wie Sie den AWS Batch Konsolenassistenten verwenden, um:

- Erstellen und konfigurieren Sie eine AWS Fargate-Rechenumgebung
- Erstellen Sie eine Job-Warteschlange.
- Erstellen Sie eine Auftragsdefinition
- Erstellen Sie einen Job und reichen Sie ihn zur Ausführung ein
- Sehen Sie sich die Ausgabe des Jobs an in CloudWatch

Erforderliche Zeit

Die Fertigstellung dieses Tutorials sollte etwa 10—15 Minuten dauern.

Regionale Einschränkungen

Es gibt keine landesspezifischen oder regionalen Einschränkungen im Zusammenhang mit der Verwendung dieser Lösung.

Kosten für die Ressourcennutzung

Für die Erstellung eines AWS Kontos fallen keine Gebühren an. Durch die Implementierung dieser Lösung können Ihnen jedoch einige oder alle Kosten entstehen, die in der folgenden Tabelle aufgeführt sind.

Beschreibung	Kosten (US-Dollar)
Die Preise basieren auf der angeforderten vCPU, dem Arbeitsspeicher, den Betriebssystemen, der CPU-Architektur und den Speicherressourcen für die Task oder den Pod.	Weitere Informationen zur Preisgestaltung finden Sie unter Fargate Pricing.

Voraussetzungen

Bevor Sie beginnen:

- Erstellen Sie eine AWS-Konto , falls Sie noch keine haben.
- Erstellen Sie die Rolle zur Aufgabenausführung. Wenn Sie die Aufgabenausführungsrolle noch nicht erstellt haben, können Sie sie im Rahmen dieses Tutorials erstellen.

Schritt 1: Erstellen Sie eine Rechenumgebung

> **Important**
>
> Um so einfach und schnell wie möglich loszulegen, enthält dieses Tutorial Schritte mit Standardeinstellungen. Wir empfehlen Ihnen, sich vor der Erstellung für den produktiven Einsatz mit allen Einstellungen vertraut zu machen und die Bereitstellung mit den Einstellungen vorzunehmen, die Ihren Anforderungen entsprechen.

Gehen Sie wie folgt vor, um eine Rechenumgebung für eine Fargate-Orchestrierung zu erstellen:

1. Öffnen Sie den Assistenten für die erste AWS Batch Ausführung der Konsole.
2. Wählen Sie für Job und Orchestrierungstyp konfigurieren die Option Fargate aus.
3. Wählen Sie Weiter aus.
4. Geben Sie im Abschnitt Konfiguration der Rechenumgebung für Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
5. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
6. Wählen Sie Weiter aus.

Schritt 2: Erstellen Sie eine Job-Warteschlange

In einer Job-Warteschlange werden Ihre eingereichten Jobs gespeichert, bis der AWS Batch Scheduler den Job auf einer Ressource in Ihrer Computerumgebung ausführt. So erstellen Sie eine Job-Warteschlange:

Gehen Sie wie folgt vor, um eine Job-Warteschlange für eine Fargate-Orchestrierung zu erstellen:

1. Geben Sie im Abschnitt Konfiguration der Job-Warteschlange für Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

2. Geben Sie als Priorität den Wert 900 für die Auftragswarteschlange ein.
3. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
4. Wählen Sie Weiter aus.

Schritt 3: Erstellen Sie eine Jobdefinition

So erstellen Sie die Jobdefinition:

1. Gehen Sie im Abschnitt Allgemeine Konfiguration wie folgt vor:
   - Geben Sie im Abschnitt Allgemeine Konfiguration für Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
2. Im Abschnitt zur Konfiguration der Fargate-Plattform:
   a. Aktivieren Sie die Option Öffentliche IP zuweisen, um eine öffentliche IP-Adresse zuzuweisen. Sie benötigen eine öffentliche IP, um das Container-Image herunterzuladen, sofern Sie kein privates Image-Repository eingerichtet haben.
   b. Wählen Sie für die Ausführungsrolle eine Aufgabenausführungsrolle aus, mit der Agenten von Amazon Elastic Container Service (Amazon ECS) in Ihrem Namen AWS Anrufe tätigen können. Wählen Sie entweder ecsTaskExecutionRolle oder BatchEcsTaskExecutionRole.

      Um die Ausführungsrolle zu erstellen, wählen Sie Ausführungsrolle erstellen. Wählen Sie im Modal „IAM-Rolle erstellen“ die Option „IAM-Rolle erstellen“.

      i. In der IAM-Konsole ist die Berechtigungseinstellung für die Erstellung der Ausführungsrolle bereits konfiguriert.
      ii. Stellen Sie sicher, dass der AWS Dienst als vertrauenswürdiger Entitätstyp ausgewählt ist.
      iii. Vergewissern Sie sich, dass für Service oder Benutzerfall Elastic Container Service ausgewählt ist.
      iv. Wählen Sie Weiter aus.
      v. Vergewissern Sie sich, dass Amazon für Berechtigungsrichtlinien ausgewählt ECSTask ExecutionRolePolicy ist.
      vi. Wählen Sie Weiter aus.

vii. Vergewissern Sie sich bei Name, review und create, dass der Rollenname lautet BatchEcsTaskExecutionRole.

viii. Wählen Sie Rolle erstellen aus.

ix. Wählen Sie in der AWS Batch Konsole die Schaltfläche „Aktualisieren“ neben „Ausführungsrolle“. Wählen Sie die BatchEcsTaskExecutionRoleAusführungsrolle aus.

3. Gehen Sie im Abschnitt Container-Konfiguration wie folgt vor:

   - Unter Command können Sie `hello world` zu einer benutzerdefinierten Nachricht wechseln oder sie unverändert lassen.

4. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
5. Wählen Sie Weiter aus.

Schritt 4: Erstellen Sie einen Job

Gehen Sie wie folgt vor, um einen Fargate-Job zu erstellen:

1. Geben Sie im Abschnitt Auftragskonfiguration für Name einen eindeutigen Namen für den Job an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
2. Für alle anderen Konfigurationsoptionen können Sie den Standardwert beibehalten.
3. Wählen Sie Weiter aus.

Schritt 5: Überprüfen und Erstellen

Überprüfen Sie auf der Seite Überprüfen und erstellen die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Ressourcen erstellen aus.

Schritt 6: Sehen Sie sich die Ausgabe des Jobs an

Gehen Sie wie folgt vor, um die Ausgabe des Jobs anzuzeigen:

1. Wählen Sie im Navigationsbereich Jobs aus.
2. Wählen Sie im Drop-down-Menü Job-Warteschlange die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.

3. In der Tabelle „Jobs“ werden alle Ihre Jobs und ihr aktueller Status aufgeführt. Sobald der Status des Job erfolgreich ist, wählen Sie den Namen des Jobs, um die Details des Jobs anzuzeigen.
4. Wählen Sie im Detailbereich den Namen des Protokollstreams aus. Die CloudWatch Konsole für den Job wird geöffnet und es sollte ein Ereignis mit der Nachricht von `hello world` oder Ihrer benutzerdefinierten Nachricht geben.

Schritt 7: Bereinigen Sie Ihre Tutorial-Ressourcen

Die EC2 Amazon-Instance wird Ihnen in Rechnung gestellt, solange sie aktiviert ist. Sie können die Instance löschen, damit keine Gebühren mehr anfallen.

Gehen Sie wie folgt vor, um die von Ihnen erstellten Ressourcen zu löschen:

1. Wählen Sie im Navigationsbereich die Option Job-Warteschlange aus.
2. Wählen Sie in der Tabelle Job-Queue die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.
3. Wählen Sie Disable (deaktivieren) aus. Sobald der Status der Job-Warteschlange deaktiviert ist, können Sie Löschen wählen.
4. Sobald die Job-Warteschlange gelöscht ist, wählen Sie im Navigationsbereich Computing-Umgebungen aus.
5. Wählen Sie die Rechenumgebung aus, die Sie für dieses Tutorial erstellt haben, und klicken Sie dann auf Deaktivieren. Es kann 1—2 Minuten dauern, bis die Deaktivierung der Rechenumgebung abgeschlossen ist.
6. Sobald der Status der Rechenumgebung Deaktiviert ist, wählen Sie Löschen aus. Es kann 1—2 Minuten dauern, bis die Rechenumgebung gelöscht ist.

Weitere Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, möchten Sie sich möglicherweise mit den folgenden Themen befassen:

- Erfahren Sie mehr über die Best Practices.
- Erkunden Sie die AWS Batch Kernkomponenten. Weitere Informationen finden Sie unter Bestandteile von AWS Batch.
- Erfahren Sie mehr über die verschiedenen verfügbaren Rechenumgebungen unter AWS Batch.

- Erfahren Sie mehr über Job-Warteschlangen und ihre verschiedenen Planungsoptionen.
- Erfahren Sie mehr über Jobdefinitionen und die verschiedenen Konfigurationsoptionen.
- Erfahren Sie mehr über die verschiedenen Arten von Jobs.

Erste Schritte mit AWS Batch und Fargate mit dem AWS CLI

Dieses Tutorial zeigt, wie Sie AWS Batch mit AWS Fargate Orchestration einen einfachen „Hello World“ -Job einrichten und mit AWS Command Line Interface ()AWS CLI ausführen. Sie erfahren, wie Sie Rechenumgebungen, Jobwarteschlangen und Jobdefinitionen erstellen und Jobs an senden. AWS Batch

Themen

- Voraussetzungen
- Erstellen Sie eine IAM-Ausführungsrolle
- Erstellen Sie eine Rechenumgebung
- Erstellen Sie eine Job-Warteschlange
- Erstellen Sie eine Auftragsdefinition
- Einen Job einreichen und überwachen
- Job-Ausgabe anzeigen
- Bereinigen von -Ressourcen
- Wir gehen zur Produktion
- Nächste Schritte

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen.

1. Das AWS CLI. Wenn Sie es installieren müssen, folgen Sie der AWS CLI Installationsanleitung. Sie können auch verwenden AWS CloudShell, was die beinhaltet AWS CLI.
2. Hat Ihre AWS CLI mit den entsprechenden Anmeldeinformationen konfiguriert. Führen Sie `aws configure` es aus, wenn Sie Ihre Anmeldeinformationen noch nicht eingerichtet haben.
3. Grundlegende Vertrautheit mit Befehlszeilenschnittstellen und Konzepten zur Containerisierung

4. Wie AWS Batch funktioniert mit IAMum AWS Batch Ressourcen, IAM-Rollen und VPC-Ressourcen in Ihrem zu erstellen und zu verwalten. AWS-Konto
5. Eine Subnetz-ID und eine Sicherheitsgruppen-ID von einer VPC in Ihrem. AWS-Konto Wenn Sie keine VPC haben, können Sie eine erstellen. Weitere Informationen zur Verwendung von AWS CLI zum Abrufen dieser Ressourcen IDs finden Sie unter describe-subnets und describe-security-groupsin der Befehlsreferenz.AWS CLI

Erforderliche Zeit: Ungefähr 15 bis 20 Minuten, um dieses Tutorial abzuschließen.

Kosten: In diesem Tutorial werden Fargate-Rechenressourcen verwendet. Die geschätzten Kosten für die Durchführung dieses Tutorials belaufen sich auf weniger als 0,01 USD, vorausgesetzt, Sie befolgen die Bereinigungsanweisungen, um Ressourcen unmittelbar nach Abschluss zu löschen. Die Fargate-Preise basieren auf den verbrauchten vCPU- und Speicherressourcen und werden pro Sekunde berechnet, mindestens jedoch 1 Minute. Aktuelle Preisinformationen finden Sie unter Preise.AWS Fargate

Erstellen Sie eine IAM-Ausführungsrolle

AWS Batch erfordert eine Ausführungsrolle, die es Agenten von Amazon Elastic Container Service (Amazon ECS) ermöglicht, AWS API-Aufrufe in Ihrem Namen durchzuführen. Diese Rolle ist für Fargate-Aufgaben erforderlich, um Container-Images abzurufen und Logs an Amazon CloudWatch zu schreiben.

Erstellen Sie ein Dokument mit Vertrauensrichtlinien

Erstellen Sie zunächst eine Vertrauensrichtlinie, die es dem Amazon ECS-Aufgabenservice ermöglicht, die Rolle zu übernehmen.

```
cat > batch-execution-role-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
EOF
```

Erstellen Sie die Ausführungsrolle

Mit dem folgenden Befehl wird eine IAM-Rolle erstellt, die `BatchEcsTaskExecutionRoleTutorial` anhand der soeben erstellten Vertrauensrichtlinie benannt wird.

```
aws iam create-role \
    --role-name BatchEcsTaskExecutionRoleTutorial \
    --assume-role-policy-document file://batch-execution-role-trust-policy.json
```

Im Folgenden wird gezeigt, wie die Ausgabe aussieht, wenn der Befehl erfolgreich ausgeführt wird.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "BatchEcsTaskExecutionRoleTutorial",
        "RoleId": "AROAUVBFO26T7xmpl3RGYTVO",
        "Arn": "arn:aws:iam::123456789012:role/BatchEcsTaskExecutionRoleTutorial",
        "CreateDate": "2025-01-13T17:34:28+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ecs-tasks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

Hängen Sie die erforderliche Richtlinie an

Fügen Sie die AWS verwaltete Richtlinie hinzu, die die erforderlichen Berechtigungen für die Ausführung von Amazon ECS-Aufgaben bereitstellt.

```
aws iam attach-role-policy \
    --role-name BatchEcsTaskExecutionRoleTutorial \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

Die Rolle kann jetzt AWS Batch für die Ausführung von Fargate-Aufgaben verwendet werden.

Erstellen Sie eine Rechenumgebung

Eine Rechenumgebung definiert die Rechenressourcen, auf denen Ihre Batch-Jobs ausgeführt werden. In diesem Tutorial erstellen Sie eine verwaltete Fargate-Rechenumgebung, die Ressourcen automatisch auf der Grundlage der Jobanforderungen bereitstellt und skaliert.

Erstellen Sie die Rechenumgebung

Der folgende Befehl erstellt eine Fargate-Rechenumgebung. Ersetzen Sie das Beispiel-Subnetz und die Sicherheitsgruppe IDs durch Ihre eigenen gemäß. Voraussetzungen

```
aws batch create-compute-environment \
    --compute-environment-name my-fargate-compute-env \
    --type MANAGED \
    --state ENABLED \
    --compute-resources type=FARGATE,maxvCpus=128,subnets=subnet-
a123456b,securityGroupIds=sg-a12b3456
```

Im Folgenden wird gezeigt, wie die Ausgabe aussieht, wenn der Befehl erfolgreich ausgeführt wird.

```
{
    "computeEnvironmentName": "my-fargate-compute-env",
    "computeEnvironmentArn": "arn:aws:batch:us-west-2:123456789012:compute-environment/
my-fargate-compute-env"
}
```

Warten Sie, bis die Rechenumgebung bereit ist

Überprüfen Sie den Status Ihrer Computerumgebung, um sicherzustellen, dass sie bereit ist, bevor Sie fortfahren.

```
aws batch describe-compute-environments \
    --compute-environments my-fargate-compute-env \
```

```
    --query 'computeEnvironments[0].status'
```

```
"VALID"
```

Wenn der Status angezeigt wird`VALID`, ist Ihre Rechenumgebung bereit, Jobs anzunehmen.

Erstellen Sie eine Job-Warteschlange

In einer Job-Warteschlange werden eingereichte Jobs gespeichert, bis der AWS Batch Scheduler sie auf Ressourcen in Ihrer Rechenumgebung ausführt. Jobs werden in der Warteschlange in der Reihenfolge ihrer Priorität verarbeitet.

Erstellen Sie die Job-Warteschlange

Der folgende Befehl erstellt eine Auftragswarteschlange mit der Priorität 900, die Ihre Fargate-Rechenumgebung verwendet.

```
aws batch create-job-queue \
    --job-queue-name my-fargate-job-queue \
    --state ENABLED \
    --priority 900 \
    --compute-environment-order order=1,computeEnvironment=my-fargate-compute-env
```

Im Folgenden wird gezeigt, wie die Ausgabe aussieht, wenn der Befehl erfolgreich ausgeführt wird.

```
{
    "jobQueueName": "my-fargate-job-queue",
    "jobQueueArn": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-
queue"
}
```

Stellen Sie sicher, dass die Auftragswarteschlange bereit ist

Vergewissern Sie sich, dass sich Ihre Auftragswarteschlange im `ENABLED` Status befindet und bereit ist, Jobs anzunehmen.

```
aws batch describe-job-queues \
    --job-queues my-fargate-job-queue \
```

```
    --query 'jobQueues[0].state' "ENABLED"
```

Erstellen Sie eine Auftragsdefinition

Eine Auftragsdefinition gibt an, wie Jobs ausgeführt werden sollen, einschließlich des zu verwendenden Docker-Images, der Ressourcenanforderungen und anderer Parameter. Für Fargate verwenden Sie Ressourcenanforderungen anstelle herkömmlicher vCPU- und Speicherparameter.

Erstellen Sie die Jobdefinition

Der folgende Befehl erstellt eine Jobdefinition, die einen einfachen „Hello World“ -Befehl unter Verwendung des Busybox-Container-Images ausführt. `123456789012`Ersetzen Sie es durch Ihre tatsächliche AWS-Konto ID und ersetzen Sie das Beispiel AWS-Region durch Ihre eigene.

```
aws batch register-job-definition \
    --job-definition-name my-fargate-job-def \
    --type container \
    --platform-capabilities FARGATE \
    --container-properties '{
        "image": "busybox",
        "resourceRequirements": [
            {"type": "VCPU", "value": "0.25"},
            {"type": "MEMORY", "value": "512"}
        ],
        "command": ["echo", "hello world"],
        "networkConfiguration": {
            "assignPublicIp": "ENABLED"
        },
        "executionRoleArn": "arn:aws:iam::123456789012:role/
BatchEcsTaskExecutionRoleTutorial"
    },
{
    "jobDefinitionName": "my-fargate-job-def",
    "jobDefinitionArn": "arn:aws:batch:us-west-2:123456789012:job-definition/my-
fargate-job-def:1",
    "revision": 1
}'
```

Die Auftragsdefinition spezifiziert 0,25 vCPU und 512 MB Arbeitsspeicher. Dies sind die Mindestressourcen für eine Fargate-Aufgabe. Die `assignPublicIp` Einstellung ist aktiviert, sodass der Container das Busybox-Image von Docker Hub abrufen kann.

Einen Job einreichen und überwachen

Da Sie nun über alle erforderlichen Komponenten verfügen, können Sie einen Job an Ihre Warteschlange senden und dessen Fortschritt überwachen.

Reichen Sie einen Job ein

Mit dem folgenden Befehl wird mithilfe der von Ihnen erstellten Auftragsdefinition ein Job an Ihre Warteschlange gesendet.

```
aws batch submit-job \
    --job-name my-hello-world-job \
    --job-queue my-fargate-job-queue \
    --job-definition my-fargate-job-def
```

Im Folgenden wird gezeigt, wie die Ausgabe aussieht, wenn der Befehl erfolgreich ausgeführt wird.

```
{
    "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
    "jobName": "my-hello-world-job",
    "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a"
}
```

Notieren Sie sich die in der Antwort `jobId` zurückgegebenen Informationen, da Sie sie verwenden werden, um den Fortschritt des Jobs zu überwachen.

Überwachen Sie den Auftragsstatus

Verwenden Sie die Job-ID, um den Status Ihres Jobs zu überprüfen. Der Job durchläuft mehrere Status:`SUBMITTED`,`PENDING`,`RUNNABLE`,`STARTING`,`RUNNING`, und schließlich `SUCCEEDED` oder`FAILED`.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Im Folgenden wird gezeigt, wie die Ausgabe aussieht, wenn der Befehl erfolgreich ausgeführt wird.

```
{
    "jobs": [
        {
```

```
            "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
            "jobName": "my-hello-world-job",
            "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a",
            "jobQueue": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-
queue",
            "status": "SUCCEEDED",
            "createdAt": 1705161908000,
            "jobDefinition": "arn:aws:batch:us-west-2:123456789012:job-definition/my-
fargate-job-def:1"
        }
    ]
}
```

Wenn der Status angezeigt wird`SUCCEEDED`, wurde Ihr Job erfolgreich abgeschlossen.

Job-Ausgabe anzeigen

Nachdem Ihr Job abgeschlossen ist, können Sie seine Ausgabe in Amazon CloudWatch Logs anzeigen.

Rufen Sie den Namen des Log-Streams ab

Rufen Sie zunächst den Namen des Log-Streams aus den Jobdetails ab. Ersetzen Sie die Beispiel-Job-ID durch Ihre eigene.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a \
    --query 'jobs[0].attempts[0].containers[0].logStreamName' \
    --output text
```

```
my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Sehen Sie sich die Jobprotokolle an

Verwenden Sie den Namen des Log-Streams, um die Ausgabe des Jobs aus CloudWatch Logs abzurufen.

```
aws logs get-log-events \
    --log-group-name /aws/batch/job \
    --log-stream-name my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a \
    --query 'events[*].message' \
```

```
    --output text
```

In der Ausgabe wird „Hello World“ angezeigt, was bestätigt, dass Ihr Job erfolgreich ausgeführt wurde.

Bereinigen von -Ressourcen

Um laufende Gebühren zu vermeiden, bereinigen Sie die Ressourcen, die Sie in diesem Tutorial erstellt haben. Aufgrund von Abhängigkeiten müssen Sie Ressourcen in der richtigen Reihenfolge löschen.

Deaktivieren und löschen Sie die Job-Warteschlange

Deaktivieren Sie zuerst die Jobwarteschlange und löschen Sie sie dann.

```
aws batch update-job-queue \
    --job-queue my-fargate-job-queue \
    --state DISABLED
```

```
aws batch delete-job-queue \
    --job-queue my-fargate-job-queue
```

Deaktivieren und löschen Sie die Rechenumgebung

Nachdem die Jobwarteschlange gelöscht wurde, deaktivieren und löschen Sie die Rechenumgebung.

```
aws batch update-compute-environment \
    --compute-environment my-fargate-compute-env \
    --state DISABLED
```

```
aws batch delete-compute-environment \
    --compute-environment my-fargate-compute-env
```

Bereinigen Sie die IAM-Rolle

Entfernen Sie den Richtlinienanhang und löschen Sie die IAM-Rolle.

```
aws iam detach-role-policy \
    --role-name BatchEcsTaskExecutionRoleTutorial \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

```
aws iam delete-role \
    --role-name BatchEcsTaskExecutionRoleTutorial
```

Temporäre Dateien entfernen

Löschen Sie die von Ihnen erstellte Vertrauensrichtlinien-Datei.

```
rm batch-execution-role-trust-policy.json
```

Alle Ressourcen wurden erfolgreich bereinigt.

Wir gehen zur Produktion

Dieses Tutorial soll Ihnen helfen zu verstehen, wie Fargate AWS Batch funktioniert. Beachten Sie bei Produktionsbereitstellungen die folgenden zusätzlichen Anforderungen:

Überlegungen zur Sicherheit:

- Erstellen Sie dedizierte Sicherheitsgruppen mit minimalem erforderlichen Zugriff, anstatt Standardsicherheitsgruppen zu verwenden
- Verwenden Sie private Subnetze mit NAT Gateway anstelle einer öffentlichen IP-Zuweisung für Container
- Speichern Sie Container-Images in Amazon ECR, anstatt öffentliche Repositorys zu verwenden
- Implementieren Sie VPC-Endpunkte für die AWS Dienstkommunikation, um Internetverkehr zu vermeiden

Überlegungen zur Architektur:

- Stellen Sie die Lösung für hohe Verfügbarkeit in mehreren Availability Zones bereit
- Implementieren Sie Strategien zur Auftragswiederholung und Warteschlangen für die Fehlerbehandlung
- Verwenden Sie mehrere Jobwarteschlangen mit unterschiedlichen Prioritäten für das Workload-Management
- Konfigurieren Sie Richtlinien für die auto Skalierung auf der Grundlage der Warteschlangentiefe und der Ressourcennutzung

- Implementieren Sie Überwachungs- und Warnmeldungen bei Auftragsausfällen und Ressourcennutzung

Betriebliche Überlegungen:

- Richten Sie CloudWatch Dashboards und Alarme für die Überwachung ein
- Implementieren Sie angemessene Protokollierungs- und Prüfpfade
- Verwenden Sie AWS CloudFormation oder die AWS CDK für die Infrastruktur als Code
- Richten Sie Sicherungs- und Notfallwiederherstellungsverfahren ein

Umfassende Anleitungen zu produktionsreifen Architekturen finden Sie im AWS Well-Architected Framework und in den Best Practices zur Sicherheit.AWS

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie sich mit weiteren Funktionen vertraut machen: AWS Batch

- Warteschlangen für Job— Erfahren Sie mehr über die Planung von Jobwarteschlangen und die Prioritätsverwaltung
- Berufsdefinitionen— Erkunden Sie erweiterte Konfigurationen für Auftragsdefinitionen, einschließlich Umgebungsvariablen, Volumes und Wiederholungsstrategien
- Rechenumgebungen für AWS Batch— Machen Sie sich mit verschiedenen Arten von Rechenumgebungen und Skalierungsoptionen vertraut
- parallel Jobs mit mehreren Knoten— Führen Sie Jobs aus, die sich über mehrere Rechenknoten erstrecken
- Ordnen Sie Jobs an— Reichen Sie eine große Anzahl ähnlicher Jobs effizient ein
- Bewährte Methoden für AWS Batch— Erlernen Sie Optimierungstechniken für Produktionsworkloads

Erste Schritte mit AWS Batch Amazon EKS

AWS Batch on Amazon EKS ist ein verwalteter Service für die Planung und Skalierung von Batch-Workloads in bestehende Amazon EKS-Cluster. AWS Batch erstellt, verwaltet oder führt keine Lebenszyklusoperationen Ihrer Amazon EKS-Cluster in Ihrem Namen durch. AWS Batch Die

Orchestrierung skaliert Knoten, die von diesen Knoten verwaltet werden, nach oben und unten AWS Batch und führt Pods auf diesen Knoten aus.

AWS Batch berührt keine Knoten, Auto Scaling-Knotengruppen oder Pod-Lebenszyklen, die nicht mit AWS Batch Rechenumgebungen in Ihrem Amazon EKS-Cluster verknüpft sind. Für AWS Batch einen effektiven Betrieb benötigt die servicebasierte Rolle Kubernetes rollenbasierte Zugriffskontrollberechtigungen (RBAC) in Ihrem vorhandenen Amazon EKS-Cluster. Weitere Informationen finden Sie in der Dokumentation unter Verwenden der RBAC-Autorisierung. Kubernetes

AWS Batch benötigt einen Kubernetes Namespace, in den Pods als AWS Batch Jobs eingebunden werden können. Wir empfehlen einen dedizierten Namespace, um die AWS Batch Pods von Ihren anderen Cluster-Workloads zu isolieren.

Nachdem AWS Batch Sie RBAC-Zugriff erhalten und ein Namespace eingerichtet wurde, können Sie diesen Amazon EKS-Cluster mithilfe der CreateComputeEnvironmentAPI-Operation einer AWS Batch Rechenumgebung zuordnen. Dieser neuen Amazon EKS-Rechenumgebung kann eine Auftragswarteschlange zugeordnet werden. AWS Batch Jobs werden auf der Grundlage einer Amazon EKS-Auftragsdefinition mithilfe des SubmitJobAPI-Vorgangs an die Auftragswarteschlange übermittelt. AWS Batch startet dann AWS Batch verwaltete Knoten und platziert Jobs aus der Job-Warteschlange als Kubernetes Pods in den EKS-Cluster, der mit einer AWS Batch Rechenumgebung verknüpft ist.

In den folgenden Abschnitten wird beschrieben, wie Sie sich AWS Batch auf Amazon EKS einrichten.

Inhalt

- Übersicht
- Voraussetzungen
- Schritt 1: Erstellen Sie Ihren Amazon EKS-Cluster für AWS Batch
- Schritt 2: Bereiten Sie Ihren Amazon EKS-Cluster vor für AWS Batch
- Schritt 3: Erstellen Sie eine Amazon EKS-Rechenumgebung
- Schritt 4: Erstellen Sie eine Job-Warteschlange und hängen Sie die Rechenumgebung an
- Schritt 5: Erstellen Sie eine Jobdefinition
- Schritt 6: Reichen Sie einen Job ein
- Schritt 7: Sehen Sie sich die Ausgabe des Jobs an
- Schritt 8: (Optional) Reichen Sie einen Job mit Überschreibungen ein

- Schritt 9: Bereinigen Sie Ihre Tutorial-Ressourcen
- Weitere Ressourcen

Übersicht

In diesem Tutorial wird gezeigt, wie Sie Amazon EKS mithilfe von AWS CLI, `kubectl` und einrichten AWS Batch `eksctl`.

Beabsichtigte Zielgruppe

Dieses Tutorial richtet sich an Systemadministratoren und Entwickler, die für die Einrichtung, das Testen und die Bereitstellung verantwortlich sind AWS Batch.

Verwendete Funktionen

Dieses Tutorial zeigt Ihnen, wie Sie das verwenden können AWS CLI, um:

- Erstellen und konfigurieren Sie eine Amazon EKS-Rechenumgebung
- Erstellen Sie eine Job-Warteschlange.
- Erstellen Sie eine Auftragsdefinition
- Erstellen Sie einen Job und reichen Sie ihn zur Ausführung ein
- Reichen Sie einen Job mit Überschreibungen ein

Erforderliche Zeit

Die Fertigstellung dieses Tutorials sollte etwa 30—40 Minuten dauern.

Regionale Einschränkungen

Es gibt keine landesspezifischen oder regionalen Einschränkungen im Zusammenhang mit der Verwendung dieser Lösung.

Kosten für die Ressourcennutzung

Für die Erstellung eines AWS Kontos fallen keine Gebühren an. Durch die Implementierung dieser Lösung können Ihnen jedoch einige oder alle Kosten entstehen, die in der folgenden Tabelle aufgeführt sind.

Beschreibung	Kosten (US-Dollar)
Die Abrechnung erfolgt pro Cluster-Stunde	Variiert je nach Instance, siehe Amazon EKS-Preise

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Tools und Ressourcen installieren und konfigurieren, die Sie benötigen, um sowohl Amazon EKS-Ressourcen als auch Amazon EKS-Ressourcen zu erstellen AWS Batch und zu verwalten.

- AWS CLI – Ein Befehlszeilen-Tool für die Arbeit mit AWS -Services, einschließlich Amazon EKS. Für dieses Handbuch müssen Sie Version 2.8.6 oder höher oder 1.26.0 oder höher verwenden. Weitere Informationen finden Sie unter Installation, Aktualisierung und Deinstallation von AWS CLI im Benutzerhandbuch.AWS Command Line Interface Nach der Installation empfehlen wir AWS CLI, dass Sie es auch konfigurieren. Weitere Informationen finden Sie unter Schnellkonfiguration mit `aws configure` im AWS Command Line Interface Benutzerhandbuch.
- **`kubectl`** – Ein Befehlszeilentool für die Arbeit mit Kubernetes-Clustern. Dieses Handbuch erfordert, dass Sie Version `1.23` oder höher verwenden. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `kubectl` im Amazon-EKS-Benutzerhandbuch.
- **`eksctl`**— Ein Befehlszeilentool für die Arbeit mit Amazon EKS-Clustern, das viele einzelne Aufgaben automatisiert. Dieses Handbuch erfordert, dass Sie Version `0.115.0` oder höher verwenden. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `eksctl` im Amazon-EKS-Benutzerhandbuch.
- Erforderliche IAM-Berechtigungen — Der von Ihnen verwendete IAM-Sicherheitsprinzipal muss über Berechtigungen für die Arbeit mit Amazon EKS-IAM-Rollen und serviceverknüpften Rollen sowie einer VPC und zugehörigen Ressourcen verfügen. AWS CloudFormation Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Elastic Kubernetes Service und Verwenden von serviceverknüpften Rollen im IAM-Benutzerhandbuch. Sie müssen alle Schritte in diesem Handbuch als derselbe Benutzer ausführen.
- Berechtigungen — Benutzer, die den CreateComputeEnvironmentAPI-Vorgang aufrufen, um eine Rechenumgebung zu erstellen, die Amazon EKS-Ressourcen verwendet, benötigen Berechtigungen für den `eks:DescribeCluster` API-Vorgang.
- AWS-Konto Nummer — Sie müssen Ihre AWS-Konto ID kennen. Folgen Sie den Anweisungen unter Ihre AWS-Konto ID anzeigen.
- (Optional) CloudWatch — Um die Details von (optional) Einen Job mit Überschreibungen einreichen zu überprüfen, muss die Protokollierung konfiguriert sein. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen.

Schritt 1: Erstellen Sie Ihren Amazon EKS-Cluster für AWS Batch

> **Important**
>
> Um so einfach und schnell wie möglich loszulegen, enthält dieses Tutorial Schritte mit Standardeinstellungen. Wir empfehlen Ihnen, sich vor der Erstellung für den produktiven Einsatz mit allen Einstellungen vertraut zu machen und die Bereitstellung mit den Einstellungen vorzunehmen, die Ihren Anforderungen entsprechen.

Nachdem Sie die Voraussetzungen installiert haben, müssen Sie Ihren Cluster mithilfe von erstellen`eksctl`. Die Erstellung des Clusters kann zwischen 10 und 15 Minuten dauern.

```
$  eksctl create cluster --name my-cluster-name --region region-code
```

Im vorherigen Befehl ersetzen Sie:

- *`my-cluster-name`*Ersetzen Sie durch den Namen, den Sie für Ihren Cluster verwenden möchten.
- *`region-code`*Ersetzen Sie beispielsweise durch AWS-Region den, in dem der Cluster erstellt werden sollus`-west-2`.

Der Clustername und die Region werden später in diesem Tutorial benötigt.

Schritt 2: Bereiten Sie Ihren Amazon EKS-Cluster vor für AWS Batch

Alle Schritte sind erforderlich.

1. Erstellen Sie einen dedizierten Namespace für Jobs AWS Batch

   Wird verwendetkubectl, um einen neuen Namespace zu erstellen.

   ```
   $ namespace=my-aws-batch-namespace
   ```

   ```
   $ cat - <<EOF | kubectl create -f -
   {
     "apiVersion": "v1",
     "kind": "Namespace",
     "metadata": {
   ```

```
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Ausgabe:

```
namespace/my-aws-batch-namespace created
```

2. Aktivieren Sie den Zugriff über die rollenbasierte Zugriffskontrolle (RBAC)

   Wird verwendet`kubectl`, um eine Kubernetes Rolle für den Cluster zu erstellen, mit der AWS Batch Knoten und Pods überwacht und die Rolle gebunden werden kann. Sie müssen dies einmal für jeden EKS-Cluster tun.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
```

```
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Ausgabe:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

3. Erstellen Sie eine Namespace-spezifische Kubernetes Rolle für die Verwaltung und den Lebenszyklus von AWS Batch Pods und deren Bindung. Sie müssen dies einmal für jeden eindeutigen Namespace tun.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
```

```
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Ausgabe:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
 created
```

4. Aktualisieren Sie Kubernetes `aws-auth` die Konfigurationsübersicht, um die vorherigen RBAC-Berechtigungen der dienstverknüpften Rolle zuzuordnen. AWS Batch

   Ersetzen Sie im folgenden Befehl:

   - Ersetze es *<your-account-number>* durch deine AWS-Konto Nummer.

```
$ eksctl create iamidentitymapping \
    --cluster my-cluster-name \
    --arn "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

   Ausgabe:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account-number>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

> **Note**
>
> Der Pfad `aws-service-role/batch.amazonaws.com/` wurde aus dem ARN der serviceverknüpften Rolle entfernt. Dies liegt an einem Problem mit der `aws-auth` Konfigurationsübersicht. Weitere Informationen finden Sie unter Rollen mit Pfaden funktionieren nicht, wenn der Pfad in ihrem ARN in der enthalten ist aws-authconfigmap.

Schritt 3: Erstellen Sie eine Amazon EKS-Rechenumgebung

AWS Batch Rechenumgebungen definieren Rechenressourcenparameter, um Ihre Batch-Workload-Anforderungen zu erfüllen. AWS Batch Hilft Ihnen in einer verwalteten Datenverarbeitungsumgebung, die Kapazität und die Instance-Typen der Rechenressourcen (KubernetesKnoten) in Ihrem Amazon EKS-Cluster zu verwalten. Dies basiert auf der Rechenressourcenspezifikation, die Sie bei der Erstellung der Rechenumgebung definieren. Sie können EC2 On-Demand-Instances oder EC2 Spot-Instances verwenden.

Jetzt, da die AWSServiceRoleForBatchserviceverknüpfte Rolle Zugriff auf Ihren Amazon EKS-Cluster hat, können Sie AWS Batch Ressourcen erstellen. Erstellen Sie zunächst eine Rechenumgebung, die auf Ihren Amazon EKS-Cluster verweist.

- Zum `subnets` Ausführen`eksctl get cluster` *`my-cluster-name`*, um die vom Cluster verwendeten Subnetze abzurufen.
- Als `securityGroupIds` Parameter können Sie dieselbe Sicherheitsgruppe wie der Amazon EKS-Cluster verwenden. Dieser Befehl ruft die Sicherheitsgruppen-ID für den Cluster ab.

```
$ aws eks describe-cluster \
    --name my-cluster-name \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Der `instanceRole` wird erstellt, wenn Sie den Cluster erstellen. Um die `instanceRole` Liste aller Entitäten zu finden, die die `AmazonEKSWorkerNodePolicy` Richtlinie verwenden:

```
$  aws iam list-entities-for-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy
```

Der Name der Richtlinienrolle enthält den Namen des Clusters, den Sie erstellt habeneksctl-*my-cluster-name*-nodegroup-example.

Führen Sie den folgenden Befehl aus, um den `instanceRole` ARN zu finden:

```
$  aws iam list-instance-profiles-for-role --role-name eksctl-my-cluster-name-
nodegroup-example
```

Ausgabe:

```
INSTANCEPROFILES        arn:aws:iam::<your-account-number>:instance-profile/
eks-04cb2200-94b9-c297-8dbe-87f12example
```

Weitere Informationen finden Sie unter Erstellen der Amazon EKS-Node-IAM-Rolle und Aktivieren des IAM-Prinzipalzugriffs auf Ihren Cluster im Amazon EKS-Benutzerhandbuch. Wenn Sie Pod-Netzwerke verwenden, finden Sie weitere Informationen unter Konfiguration des Amazon VPC Kubernetes CNI-Plug-ins für die Verwendung von IAM-Rollen für Dienstkonten im Amazon EKS-Benutzerhandbuch.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:region-code:your-account-number:cluster/my-cluster-
name",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
```

```
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-environment.json
```

Hinweise

- Die Wartung einer Amazon EKS-Rechenumgebung ist eine gemeinsame Verantwortung. Weitere Informationen finden Sie unter Gemeinsame Verantwortung der Kubernetes Knoten.

Schritt 4: Erstellen Sie eine Job-Warteschlange und hängen Sie die Rechenumgebung an

> **Important**
>
> Bevor Sie fortfahren, müssen Sie sich vergewissern, dass die Rechenumgebung fehlerfrei ist. Dazu kann die DescribeComputeEnvironmentsAPI-Operation verwendet werden.
>
> ```
> $ aws batch describe-compute-environments --compute-environments My-Eks-CE1
> ```
>
> Vergewissern Sie sich, dass der `status` Parameter dies nicht ist`INVALID`. Wenn ja, schauen Sie sich den `statusReason` Parameter für die Ursache an. Weitere Informationen finden Sie unter Problembehebung AWS Batch.

Jobs, die an diese neue Auftragswarteschlange gesendet wurden, werden als Pods auf AWS Batch verwalteten Knoten ausgeführt, die dem Amazon EKS-Cluster beigetreten sind, der mit Ihrer Rechenumgebung verknüpft ist.

```
$ cat <<EOF > ./batch-eks-job-queue.json
 {
    "jobQueueName": "My-Eks-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-CE1"
      }
    ]
  }
EOF
```

```
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Schritt 5: Erstellen Sie eine Jobdefinition

Die folgende Jobdefinition weist den Pod an, 60 Sekunden lang in den Ruhemodus zu wechseln.

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
```

```
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Hinweise

- Es gibt Überlegungen zu den `memory` Parametern `cpu` und. Weitere Informationen finden Sie unter Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS.

Schritt 6: Reichen Sie einen Job ein

Führen Sie den folgenden AWS CLI Befehl aus, um einen neuen Job einzureichen.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1
```

Um den Status eines Job zu überprüfen:

```
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Hinweise

- Weitere Informationen zum Ausführen von Aufträgen auf Amazon EKS-Ressourcen finden Sie unterStellenangebote bei Amazon EKS.

Schritt 7: Sehen Sie sich die Ausgabe des Jobs an

Gehen Sie wie folgt vor, um die Ausgabe des Jobs anzuzeigen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.

2. Wählen Sie im Navigationsbereich Jobs aus.
3. Wählen Sie im Drop-down-Menü Job-Warteschlange die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.
4. In der Tabelle „Jobs“ werden alle Ihre Jobs und ihr aktueller Status aufgeführt. Sobald der Status des Job erfolgreich ist, wählen Sie den Namen des Jobs*My-Eks-JQ1*, um die Details des Jobs anzuzeigen.
5. Im Detailbereich sollten die Zeitpunkte „Gestartet am“ und „Beendet zu“ jeweils eine Minute voneinander entfernt sein.

Schritt 8: (Optional) Reichen Sie einen Job mit Überschreibungen ein

Dieser Job überschreibt den an den Container übergebenen Befehl. AWS Batch säubert die Pods nach Abschluss der Jobs aggressiv, um die Last auf zu reduzieren. Kubernetes Um die Details eines Auftrags überprüfen zu können, muss die Protokollierung konfiguriert werden. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
```

```
$ aws batch submit-job --cli-input-json file://./submit-job-override.json
```

Hinweise

- Aktivieren Sie die Protokollierung der Amazon EKS-Kontrollebene, um einen besseren Einblick in die Einzelheiten der Vorgänge zu erhalten. Weitere Informationen finden Sie unter Protokollierung der Amazon EKS-Kontrollebene im Amazon EKS-Benutzerhandbuch.
- Daemonsetsund der kubelets Overhead wirkt sich auf die verfügbaren vCPU- und Speicherressourcen aus, insbesondere auf Skalierung und Stellenvermittlung. Weitere Informationen finden Sie unter Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS.

Gehen Sie wie folgt vor, um die Ausgabe des Jobs anzuzeigen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie im Navigationsbereich Jobs aus.
3. Wählen Sie im Drop-down-Menü Job-Warteschlange die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.
4. In der Tabelle „Jobs“ werden alle Ihre Jobs und ihr aktueller Status aufgeführt. Sobald der Status des Job erfolgreich ist, wählen Sie den Namen des Jobs, um die Details des Jobs anzuzeigen.
5. Wählen Sie im Detailbereich den Namen des Protokollstreams aus. Die CloudWatch Konsole für den Job wird geöffnet und es sollte ein Ereignis mit der Nachricht von `hello world` oder Ihrer benutzerdefinierten Nachricht geben.

Schritt 9: Bereinigen Sie Ihre Tutorial-Ressourcen

Die EC2 Amazon-Instance wird Ihnen in Rechnung gestellt, solange sie aktiviert ist. Sie können die Instance löschen, damit keine Gebühren mehr anfallen.

Gehen Sie wie folgt vor, um die von Ihnen erstellten Ressourcen zu löschen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie im Navigationsbereich die Option Job-Warteschlange aus.
3. Wählen Sie in der Tabelle Job-Queue die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.

4. Wählen Sie Disable (deaktivieren) aus. Sobald der Status der Job-Warteschlange deaktiviert ist, können Sie Löschen wählen.
5. Sobald die Job-Warteschlange gelöscht ist, wählen Sie im Navigationsbereich Computing-Umgebungen aus.
6. Wählen Sie die Rechenumgebung aus, die Sie für dieses Tutorial erstellt haben, und klicken Sie dann auf Deaktivieren. Es kann 1—2 Minuten dauern, bis die Deaktivierung der Rechenumgebung abgeschlossen ist.
7. Sobald der Status der Rechenumgebung Deaktiviert ist, wählen Sie Löschen aus. Es kann 1—2 Minuten dauern, bis die Rechenumgebung gelöscht ist.

Weitere Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, möchten Sie sich möglicherweise mit den folgenden Themen befassen:

- Erfahren Sie mehr über die Best Practices.
- Erkunden Sie die AWS Batch Kernkomponenten. Weitere Informationen finden Sie unter Bestandteile von AWS Batch.
- Erfahren Sie mehr über die verschiedenen verfügbaren Rechenumgebungen unter AWS Batch.
- Erfahren Sie mehr über Job-Warteschlangen und ihre verschiedenen Planungsoptionen.
- Erfahren Sie mehr über Jobdefinitionen und die verschiedenen Konfigurationsoptionen.
- Erfahren Sie mehr über die verschiedenen Arten von Jobs.

Erste Schritte mit AWS Batch Amazon EKS Private Clusters

AWS Batch ist ein verwalteter Service, der Batch-Workloads in Ihren Amazon Elastic Kubernetes Service (Amazon EKS) -Clustern orchestriert. Dazu gehören Warteschlangen, Abhängigkeitsverfolgung, verwaltete Auftragswiederholungen und Prioritäten, Pod-Verwaltung und Knotenskalierung. Diese Funktion verbindet Ihren bestehenden privaten Amazon EKS-Cluster mit AWS Batch , um Ihre Jobs skalierbar auszuführen. Sie können `eksctl`(eine Befehlszeilenschnittstelle für Amazon EKS), die AWS Konsole oder die verwenden, AWS Command Line Interfaceum einen privaten Amazon EKS-Cluster mit allen anderen erforderlichen Ressourcen zu erstellen.

Amazon EKS-Cluster, die nur privat sind, haben standardmäßig keinen inbound/outbound Internetzugang, und Sie können nur von Ihrer VPC oder einem verbundenen Netzwerk aus auf den API-Server zugreifen. Amazon VPC-Endpunkte werden verwendet, um privaten Zugriff auf andere AWS Dienste zu ermöglichen. `eksctl`unterstützt die Erstellung vollständig privater Cluster mit einer bereits vorhandenen Amazon VPC und Subnetzen. `eksctl`erstellt auch Amazon VPC-Endpunkte in der bereitgestellten Amazon VPC und ändert Routentabellen für die bereitgestellten Subnetze.

Jedem Subnetz sollte eine explizite Routing-Tabelle zugeordnet sein, da die Haupt-Routing-Tabelle `eksctl` nicht verändert wird. Ihr Cluster muss Images aus einer Container-Registry abrufen, die sich in Ihrer Amazon VPC befindet. Sie können auch eine Amazon Elastic Container Registry in Ihrer Amazon VPC erstellen und Container-Images dorthin kopieren, damit Ihre Knoten sie abrufen können. Weitere Informationen finden Sie unter Kopieren eines Container-Images von einem Repository in ein anderes Repository. Informationen zu den ersten Schritten mit privaten Amazon ECR-Repositorys finden Sie unter Private Amazon ECR-Repositorys.

Sie können optional eine Pull-Through-Cache-Regel mit Amazon ECR erstellen. Sobald eine Pull-Through-Cache-Regel für eine externe öffentliche Registry erstellt wurde, können Sie mithilfe Ihrer Amazon ECR Private Registry Uniform Resource Identifier (URI) ein Image aus dieser externen öffentlichen Registrierung abrufen. Dann erstellt Amazon ECR ein Repository und speichert das Bild im Cache. Wenn ein zwischengespeichertes Image mithilfe der privaten Amazon ECR-Registrierungs-URI abgerufen wird, überprüft Amazon ECR die Remote-Registrierung, um festzustellen, ob es eine neue Version des Images gibt, und aktualisiert Ihre private Registrierung bis zu einmal alle 24 Stunden.

Inhalt

- Übersicht
- Voraussetzungen
- Schritt 1: Erstellen Sie Ihren EKS-Cluster für AWS Batch
- Schritt 2: Bereiten Sie Ihren EKS-Cluster vor für AWS Batch
- Schritt 3: Erstellen Sie eine Amazon EKS-Rechenumgebung
- Schritt 4: Erstellen Sie eine Job-Warteschlange und hängen Sie die Rechenumgebung an
- Schritt 5: Erstellen Sie einen Amazon ECR mit Pull-Through-Cache
- Schritt 6: Registrieren Sie eine Jobdefinition
- Schritt 7: Reichen Sie einen Job zur Ausführung ein
- Schritt 8: Sehen Sie sich die Ausgabe des Jobs an

- Schritt 9: (Optional) Reichen Sie einen Job mit Überschreibungen ein
- Schritt 10: Bereinigen Sie Ihre Tutorial-Ressourcen
- Weitere Ressourcen
- Fehlerbehebung

Übersicht

Dieses Tutorial zeigt, wie Sie AWS Batch mit einem privaten Amazon EKS mithilfe von AWS CloudShell, `kubectl` und einrichten`eksctl`.

Beabsichtigte Zielgruppe

Dieses Tutorial richtet sich an Systemadministratoren und Entwickler, die für die Einrichtung, das Testen und die Bereitstellung verantwortlich sind AWS Batch.

Verwendete Funktionen

Dieses Tutorial zeigt Ihnen, wie Sie das verwenden AWS CLI, um:

- Verwenden Sie Amazon Elastic Container Registry (Amazon ECR) zum Speichern von Container-Images
- Erstellen und konfigurieren Sie eine Amazon EKS-Rechenumgebung
- Erstellen Sie eine Job-Warteschlange.
- Erstellen Sie eine Auftragsdefinition
- Erstellen Sie einen Job und reichen Sie ihn zur Ausführung ein
- Reichen Sie einen Job mit Überschreibungen ein

Erforderliche Zeit

Die Fertigstellung dieses Tutorials sollte etwa 40—50 Minuten dauern.

Regionale Einschränkungen

Es gibt keine landesspezifischen oder regionalen Einschränkungen im Zusammenhang mit der Verwendung dieser Lösung.

Kosten für die Ressourcennutzung

Für die Erstellung eines AWS Kontos fallen keine Gebühren an. Durch die Implementierung dieser Lösung können Ihnen jedoch einige oder alle Kosten entstehen, die in der folgenden Tabelle aufgeführt sind.

Beschreibung	Kosten (US-Dollar)
Die Abrechnung erfolgt pro Cluster-Stunde	Variiert je nach Instance, siehe Amazon EKS-Preise
EC2 Amazon-Instanz	Sie zahlen für jede EC2 Amazon-Instance, die erstellt wird. Weitere Informationen zur Preisgestaltung finden Sie unter EC2 Amazon-Preise.

Voraussetzungen

In diesem Tutorial wird eine browserbasierte, vorauthentifizierte Shell verwendet AWS CloudShell , die Sie direkt von der aus starten. AWS Management Console Dies ermöglicht den Zugriff auf den Cluster, sobald dieser keinen öffentlichen Internetzugang mehr hat. Das AWS CLI`kubectl`, und ist `eksctl` möglicherweise bereits als Teil von installiert AWS CloudShell. Weitere Informationen zu AWS CloudShell finden Sie im AWS CloudShellBenutzerhandbuch. Eine Alternative dazu AWS CloudShell besteht darin, eine Verbindung zur VPC Ihres Clusters oder zu einem verbundenen Netzwerk herzustellen.

Um kubectl-Befehle auszuführen, benötigen Sie privaten Zugriff auf Ihren Amazon EKS-Cluster. Das bedeutet, dass der gesamte Datenverkehr zu Ihrem Cluster-API-Server aus der VPC Ihres Clusters oder einem verbundenen Netzwerk stammen muss.

- AWS CLI— Ein Befehlszeilentool für die Arbeit mit AWS Diensten, einschließlich Amazon EKS. Für dieses Handbuch müssen Sie Version 2.8.6 oder höher oder 1.26.0 oder höher verwenden. Weitere Informationen finden Sie unter Installation, Aktualisierung und Deinstallation von AWS CLI im Benutzerhandbuch.AWS Command Line Interface Nach der Installation empfehlen wir AWS CLI, dass Sie es auch konfigurieren. Weitere Informationen finden Sie unter Schnellkonfiguration mit `aws configure` im AWS Command Line Interface Benutzerhandbuch.
- **`kubectl`** – Ein Befehlszeilentool für die Arbeit mit Kubernetes-Clustern. Dieses Handbuch erfordert, dass Sie Version `1.23` oder höher verwenden. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `kubectl` im Amazon-EKS-Benutzerhandbuch.
- **`eksctl`**— Ein Befehlszeilentool für die Arbeit mit Amazon EKS-Clustern, das viele einzelne Aufgaben automatisiert. Dieses Handbuch erfordert, dass Sie Version `0.115.0` oder höher

verwenden. Weitere Informationen finden Sie unter Installieren oder Aktualisieren von `eksctl` im Amazon-EKS-Benutzerhandbuch.

- Berechtigungen — Benutzer, die den CreateComputeEnvironmentAPI-Vorgang aufrufen, um eine Rechenumgebung zu erstellen, die Amazon EKS-Ressourcen verwendet, benötigen Berechtigungen für den `eks:DescribeCluster` und `eks:ListClusters` API-Vorgang. Sie können die AWSBatchFullAccessverwaltete Richtlinie an Ihr Benutzerkonto anhängen, indem Sie den Anweisungen zum Hinzufügen und Entfernen von IAM-Identitätsberechtigungen im IAM-Benutzerhandbuch folgen.
- InstanceRole— Sie müssen `InstanceRole` für Ihre Amazon EKS-Knoten einen erstellen, der die `AmazonEC2ContainerRegistryPullOnly` Richtlinien `AmazonEKSWorkerNodePolicy` und enthält. Anweisungen zur Erstellung der IAM-Rolle für Amazon EKS Node `InstanceRole` finden Sie unter Creating the Amazon EKS node. Sie benötigen den ARN von`InstanceRole`.
- AWS-Konto ID — Sie müssen Ihre AWS-Konto ID kennen. Folgen Sie den Anweisungen unter Ihre AWS-Konto ID anzeigen.
- (Optional) CloudWatch — Um die Details von (optional) Einen Job mit Überschreibungen einreichen zu überprüfen, muss die Protokollierung konfiguriert sein. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen.

Schritt 1: Erstellen Sie Ihren EKS-Cluster für AWS Batch

> **Important**
>
> Um so einfach und schnell wie möglich loszulegen, enthält dieses Tutorial Schritte mit Standardeinstellungen. Wir empfehlen Ihnen, sich vor der Erstellung für den produktiven Einsatz mit allen Einstellungen vertraut zu machen und die Bereitstellung mit den Einstellungen vorzunehmen, die Ihren Anforderungen entsprechen.

Wir empfehlen Ihnen, die folgende Konfigurationsdatei zu verwenden`eksctl`, um Ihren Cluster zu erstellen. Um Ihren Cluster manuell einzurichten, folgen Sie den Anweisungen unter Bereitstellen von privaten Clustern mit eingeschränktem Internetzugang im Amazon EKS-Benutzerhandbuch.

1. Öffnen Sie die AWS CloudShell Konsole und stellen Sie die Region auf ein`us-east-1`. Stellen Sie für den Rest des Tutorials sicher, dass Sie verwenden`us-east-1`.
2. Erstellen Sie mithilfe der `eksctl` Beispielkonfigurationsdatei einen privaten EKS-Cluster in der `us-east-1` Region. Speichern Sie die Yaml-Datei in Ihrer AWS CloudShell Umgebung und

geben Sie ihr einen Namen. `clusterConfig.yaml` Sie können den Namen ändern*my-test-cluster*, den Sie für Ihren Cluster verwenden möchten.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
metadata:
    name: my-test-cluster
    region: us-east-1
availabilityZones:
    - us-east-1a
    - us-east-1b
    - us-east-1c
managedNodeGroups:
    - name: ng-1
      privateNetworking: true
privateCluster:
    enabled: true
    skipEndpointCreation: false
```

3. Erstellen Sie Ihre Ressourcen mit dem folgenden Befehl:. `eksctl create cluster -f clusterConfig.yaml` Die Clustererstellung kann zwischen 10 und 15 Minuten dauern.
4. Sobald die Erstellung des Clusters abgeschlossen ist, müssen Sie Ihre AWS CloudShell IP-Adresse zur Zulassungsliste hinzufügen. Um Ihre AWS CloudShell IP-Adresse zu finden, führen Sie den folgenden Befehl aus:

```
curl http://checkip.amazonaws.com
```

Sobald Sie die öffentliche IP-Adresse haben, müssen Sie eine Regel für die Zulassungsliste erstellen:

```
aws eks update-cluster-config \
  --name my-test-cluster \
  --region us-east-1 \
  --resources-vpc-config
 endpointPublicAccess=true,endpointPrivateAccess=true,publicAccessCidrs=["<Public
 IP>/32"]
```

Wenden Sie dann das Update auf die kubectl-Konfigurationsdatei an:

```
aws eks update-kubeconfig --name my-test-cluster --region us-east-1
```

5. Um zu testen, ob Sie Zugriff auf die Knoten haben, führen Sie den folgenden Befehl aus:

```
kubectl get nodes
```

Die Ausgabe des Befehls lautet:

```
NAME                                STATUS   ROLES    AGE     VERSION
ip-192-168-107-235.ec2.internal     Ready    none     1h   v1.32.3-eks-473151a
ip-192-168-165-40.ec2.internal      Ready    none     1h   v1.32.3-eks-473151a
ip-192-168-98-54.ec2.internal       Ready    none     1h   v1.32.1-eks-5d632ec
```

Schritt 2: Bereiten Sie Ihren EKS-Cluster vor für AWS Batch

Alle Schritte sind erforderlich und müssen in ausgeführt werden AWS CloudShell.

1. Erstellen Sie einen dedizierten Namespace für Jobs AWS Batch

   Wird verwendet`kubectl`, um einen neuen Namespace zu erstellen.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

   Ausgabe:

```
namespace/my-aws-batch-namespace created
```

2. Aktivieren Sie den Zugriff über die rollenbasierte Zugriffskontrolle (RBAC)

   Wird verwendet`kubectl`, um eine Kubernetes Rolle für den Cluster zu erstellen, mit der AWS Batch Knoten und Pods überwacht und die Rolle gebunden werden kann. Sie müssen dies einmal für jeden Amazon EKS-Cluster tun.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
```

```
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Ausgabe:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Erstellen Sie eine Namespace-spezifische Kubernetes Rolle für die Verwaltung und den Lebenszyklus von Pods und AWS Batch binden Sie sie. Sie müssen dies einmal für jeden eindeutigen Namespace tun.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
```

```
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Ausgabe:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
 created
```

Aktualisieren Sie Kubernetes `aws-auth` die Konfigurationsübersicht, um die vorherigen RBAC-Berechtigungen der dienstverknüpften Rolle zuzuordnen. AWS Batch

```
$ eksctl create iamidentitymapping \
    --cluster my-test-cluster \
    --arn "arn:aws:iam::<your-account-ID>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

Ausgabe:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account-ID>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

> **Note**
>
> Der Pfad `aws-service-role/batch.amazonaws.com/` wurde aus dem ARN der serviceverknüpften Rolle entfernt. Dies liegt an einem Problem mit der `aws-auth` Konfigurationsübersicht. Weitere Informationen finden Sie unter Rollen mit Pfaden funktionieren nicht, wenn der Pfad in ihrem ARN in der enthalten ist aws-authconfigmap.

Schritt 3: Erstellen Sie eine Amazon EKS-Rechenumgebung

AWS Batch Rechenumgebungen definieren Rechenressourcenparameter, um Ihre Batch-Workload-Anforderungen zu erfüllen. AWS Batch Hilft Ihnen in einer verwalteten Rechenumgebung, die Kapazität und die Instance-Typen der Rechenressourcen (KubernetesKnoten) in Ihrem Amazon EKS-

Cluster zu verwalten. Dies basiert auf der Rechenressourcenspezifikation, die Sie bei der Erstellung der Rechenumgebung definieren. Sie können EC2 On-Demand-Instances oder EC2 Spot-Instances verwenden.

Jetzt, da die AWSServiceRoleForBatchserviceverknüpfte Rolle Zugriff auf Ihren Amazon EKS-Cluster hat, können Sie AWS Batch Ressourcen erstellen. Erstellen Sie zunächst eine Rechenumgebung, die auf Ihren Amazon EKS-Cluster verweist.

- Zum `subnets` Ausführen`eksctl get cluster` *`my-test-cluster`*, um die vom Cluster verwendeten Subnetze abzurufen.
- Als `securityGroupIds` Parameter können Sie dieselbe Sicherheitsgruppe wie der Amazon EKS-Cluster verwenden. Dieser Befehl ruft die Sicherheitsgruppen-ID für den Cluster ab.

```
$ aws eks describe-cluster \
    --name my-test-cluster \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Verwenden Sie den ARN, den `instanceRole` Sie in den Voraussetzungen erstellt haben.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:us-east-1:<your-account-ID>:cluster/my-test-cluster",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
```

```
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-environment.json
```

Hinweise

- Die Wartung einer Amazon EKS-Rechenumgebung ist eine gemeinsame Verantwortung. Weitere Informationen finden Sie unter Sicherheit in Amazon EKS.

Schritt 4: Erstellen Sie eine Job-Warteschlange und hängen Sie die Rechenumgebung an

> **Important**
>
> Bevor Sie fortfahren, müssen Sie sich vergewissern, dass die Rechenumgebung fehlerfrei ist. Dazu kann die DescribeComputeEnvironmentsAPI-Operation verwendet werden.
>
> ```
> $ aws batch describe-compute-environments --compute-environments My-Eks-CE1
> ```
>
> Vergewissern Sie sich, dass der `status` Parameter dies nicht ist`INVALID`. Wenn ja, schauen Sie sich den `statusReason` Parameter für die Ursache an. Weitere Informationen finden Sie unter Problembehebung AWS Batch.

Jobs, die an diese neue Auftragswarteschlange gesendet wurden, werden als Pods auf AWS Batch verwalteten Knoten ausgeführt, die dem Amazon EKS-Cluster beigetreten sind, der mit Ihrer Rechenumgebung verknüpft ist.

```
$ cat <<EOF > ./batch-eks-job-queue.json
 {
    "jobQueueName": "My-Eks-JQ1",
    "priority": 10,
```

```
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-CE1"
      }
    ]
  }
EOF
```

```
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Schritt 5: Erstellen Sie einen Amazon ECR mit Pull-Through-Cache

Da der Cluster keinen öffentlichen Internetzugang hat, müssen Sie einen Amazon ECR für Container-Images erstellen. Mit den folgenden Anweisungen wird ein Amazon ECR mit einer Pull-Through-Cache-Regel zum Speichern des Images erstellt.

1. Mit dem folgenden Befehl wird die Pull-Through-Cache-Regel erstellt. Sie können es durch ein anderes *tutorial-prefix* Präfix ersetzen.

   ```
   aws ecr create-pull-through-cache-rule \
       --ecr-repository-prefix "my-prefix" \
       --upstream-registry-url "public.ecr.aws" \
       --region us-east-1
   ```

2. Authentifizieren Sie sich mit dem öffentlichen ECR.

   ```
   aws ecr get-login-password --region us-east-1 | docker login --username AWS --
   password-stdin <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com
   ```

   Jetzt können Sie ein Bild abrufen.

   ```
   docker pull <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/
   amazonlinux/amazonlinux:2
   ```

3. Sie können das Repository und das Image überprüfen, indem Sie die folgenden Befehle ausführen:

   ```
   aws ecr describe-repositories
   ```

```
aws ecr describe-images --repository-name my-prefix/amazonlinux/amazonlinux
```

4. Die Bildzeichenfolge, die zum Abrufen des Containers verwendet werden soll, hat das folgende Format:

```
<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/amazonlinux/
amazonlinux:2
```

Schritt 6: Registrieren Sie eine Jobdefinition

Die folgende Jobdefinition weist den Pod an, 60 Sekunden lang in den Ruhemodus zu wechseln.

Anstatt einen Link zu einem Bild in einem öffentlichen ECR-Repository bereitzustellen, geben Sie im Bildfeld der Jobdefinition den Link zu dem Bild ein, das in unserem privaten ECR-Repository gespeichert ist. Sehen Sie sich die folgende Beispiel-Auftragsdefinition an:

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/
amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
```

```
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Hinweise

- Es gibt Überlegungen zu den `memory` Parametern `cpu` und. Weitere Informationen finden Sie unter Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS.

Schritt 7: Reichen Sie einen Job zur Ausführung ein

Führen Sie den folgenden AWS CLI Befehl aus AWS CloudShell , um einen neuen Job einzureichen, und geben Sie die eindeutige JobID zurück.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1
```

Hinweise

- Weitere Informationen zum Ausführen von Aufträgen auf Amazon EKS-Ressourcen finden Sie unterStellenangebote bei Amazon EKS.

Schritt 8: Sehen Sie sich die Ausgabe des Jobs an

Um den Status eines Job zu überprüfen:

```
$ aws batch describe-jobs --job <JobID-from-submit-response>
```

Das `startedAt` und `stoppedAt` sollte eine Minute voneinander entfernt sein.

Schritt 9: (Optional) Reichen Sie einen Job mit Überschreibungen ein

Dieser Job überschreibt den an den Container übergebenen Befehl.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
```

```
$ aws batch submit-job - -cli-input-json file://./submit-job-override.json
```

Hinweise

- Aktivieren Sie die Protokollierung der Amazon EKS-Kontrollebene, um einen besseren Einblick in die Einzelheiten der Vorgänge zu erhalten. Weitere Informationen finden Sie unter Protokollierung der Amazon EKS-Kontrollebene im Amazon EKS-Benutzerhandbuch.
- Daemonsetsund der kubelets Overhead wirkt sich auf die verfügbaren vCPU- und Speicherressourcen aus, insbesondere auf Skalierung und Stellenvermittlung. Weitere Informationen finden Sie unter Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS.

Schritt 10: Bereinigen Sie Ihre Tutorial-Ressourcen

Die EC2 Amazon-Instance wird Ihnen in Rechnung gestellt, solange sie aktiviert ist. Sie können die Instance löschen, damit keine Gebühren mehr anfallen.

Gehen Sie wie folgt vor, um die von Ihnen erstellten Ressourcen zu löschen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie im Navigationsbereich die Option Job-Warteschlange aus.
3. Wählen Sie in der Tabelle Job-Queue die Job-Warteschlange aus, die Sie für das Tutorial erstellt haben.
4. Wählen Sie unter Aktionen die Option Deaktivieren aus. Sobald der Status der Job-Warteschlange deaktiviert ist, können Sie Löschen wählen.
5. Sobald die Job-Warteschlange gelöscht ist, wählen Sie im Navigationsbereich Computing-Umgebungen aus.
6. Wählen Sie die Rechenumgebung aus, die Sie für dieses Tutorial erstellt haben, und wählen Sie dann unter Aktionen die Option Deaktivieren aus. Es kann 1—2 Minuten dauern, bis die Deaktivierung der Rechenumgebung abgeschlossen ist.
7. Sobald der Status der Rechenumgebung Deaktiviert ist, wählen Sie Löschen aus. Es kann 1—2 Minuten dauern, bis die Rechenumgebung gelöscht ist.

Weitere Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, möchten Sie sich möglicherweise mit den folgenden Themen befassen:

- Erfahren Sie mehr über die Best Practices.
- Erkunden Sie die AWS Batch Kernkomponenten. Weitere Informationen finden Sie unter Bestandteile von AWS Batch.
- Erfahren Sie mehr über die verschiedenen verfügbaren Rechenumgebungen unter AWS Batch.
- Erfahren Sie mehr über Job-Warteschlangen und ihre verschiedenen Planungsoptionen.
- Erfahren Sie mehr über Jobdefinitionen und die verschiedenen Konfigurationsoptionen.
- Erfahren Sie mehr über die verschiedenen Arten von Jobs.

Fehlerbehebung

Wenn Knoten, die von gestartet wurden, AWS Batch keinen Zugriff auf das Amazon ECR-Repository (oder ein anderes Repository) haben, in dem Ihr Image gespeichert ist, können Ihre Jobs im Status STARTING verbleiben. Das liegt daran, dass der Pod das Image nicht herunterladen und Ihren AWS Batch Job ausführen kann. Wenn Sie auf den von gestarteten Pod-Namen klicken, sollten AWS Batch Sie die Fehlermeldung sehen und das Problem bestätigen können. Die Fehlermeldung sollte etwa wie folgt aussehen:

```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =
Unknown desc = failed to pull and unpack image
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Weitere häufig auftretende Problembehandlungsszenarien finden Sie unter Problembehandlung AWS Batch. Informationen zur Fehlerbehebung basierend auf dem Pod-Status finden Sie unter Wie behebe ich Probleme mit dem Pod-Status in Amazon EKS? .

Das AWS Batch Widget-Dashboard

Über das AWS Batch Dashboard können Sie aktuelle Jobs, Jobwarteschlangen und Rechenumgebungen überwachen. Standardmäßig werden die folgenden Dashboard-Widgets angezeigt:

- Jobübersicht — Weitere Informationen zu AWS Batch Jobs finden Sie unterAufträge.
- Übersicht über die Auftragswarteschleife — Weitere Informationen zu AWS Batch Auftragswarteschlangen finden Sie unterWarteschlangen für Job.
- Überblick über die Rechenumgebung — Weitere Informationen zu AWS Batch Rechenumgebungen finden Sie unterRechenumgebungen für AWS Batch.

Sie können die Widgets anpassen, die auf der Dashboard-Seite angezeigt werden. In den folgenden Abschnitten werden zusätzliche Widgets beschrieben, die Sie hinzufügen können.

Themen

- Wie füge ich das Widget „Single Job Queue“ zum AWS Batch Dashboard hinzu
- Wie füge ich das CloudWatch Container Insights-Widget zum AWS Batch Dashboard hinzu
- So fügen Sie das Job-Logs-Widget zum AWS Batch Dashboard hinzu

Wie füge ich das Widget „Single Job Queue“ zum AWS Batch Dashboard hinzu

Das Widget „Einzelne Auftragswarteschlange“ zeigt detaillierte Informationen zu einer einzelnen Auftragswarteschlange an.

Gehen Sie wie folgt vor, um dieses Widget hinzuzufügen.

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie in der Navigationsleiste AWS-Region das gewünschte aus.
3. Wählen Sie im Navigationsbereich Dashboard (Dashboard).
4. Wählen Sie Widgets hinzufügen aus.
5. Wählen Sie unter Warteschlange für einzelne Jobs die Option Widget hinzufügen aus.
6. Wählen Sie Job Job-Warteschlange die gewünschte Job-Warteschlange aus.

7. Wählen Sie unter Jobstatus die Job-Status aus, die Sie anzeigen möchten.
8. (Optional) Deaktivieren Sie die Option Verbundene Rechenumgebungen anzeigen, wenn Sie die Eigenschaften für Rechenumgebungen nicht anzeigen möchten.
9. Wählen Sie unter Eigenschaften der Rechenumgebung die gewünschten Eigenschaften aus.
10. Wählen Sie Hinzufügen aus.

Wie füge ich das CloudWatch Container Insights-Widget zum AWS Batch Dashboard hinzu

Dieses Widget zeigt aggregierte Metriken für AWS Batch Rechenumgebungen und Jobs an. Weitere Informationen zu Container Insights finden Sie unter the section called “CloudWatch Einblicke in Container”.

Gehen Sie wie folgt vor, um dieses Widget hinzuzufügen.

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie in der Navigationsleiste AWS-Region das gewünschte aus.
3. Wählen Sie im Navigationsbereich Dashboard (Dashboard).
4. Wählen Sie Widgets hinzufügen aus.
5. Wählen Sie für Container-Einblicke die Option Widget hinzufügen aus.
6. Wählen Sie unter Rechenumgebung die gewünschte Rechenumgebung aus.
7. Wählen Sie Hinzufügen aus.

So fügen Sie das Job-Logs-Widget zum AWS Batch Dashboard hinzu

Dieses Widget zeigt verschiedene Protokolle Ihrer Jobs an einem übersichtlichen Ort an. Weitere Informationen zu Job-Logs finden Sie unterthe section called “Tutorial: AWS Batch Jobprotokolle an CloudWatch Logs senden”.

Gehen Sie wie folgt vor, um dieses Widget hinzuzufügen.

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie in der Navigationsleiste AWS-Region das gewünschte aus.

3. Wählen Sie im Navigationsbereich Dashboard (Dashboard).
4. Wählen Sie Widgets hinzufügen aus.
5. Wählen Sie für Job-Logs die Option Widget hinzufügen.
6. Geben Sie unter Job-ID die Job-ID für den gewünschten Job ein.
7. Wählen Sie Hinzufügen aus.

Rechenumgebungen für AWS Batch

Auftragswarteschlangen werden einer oder mehreren Datenverarbeitungsumgebungen zugeordnet. Rechenumgebungen enthalten die Amazon ECS-Container-Instances, die zur Ausführung von containerisierten Batch-Jobs verwendet werden. Eine bestimmte Rechenumgebung kann auch einer oder mehreren Auftragswarteschlangen zugeordnet werden. Innerhalb einer Job-Warteschlange haben die zugehörigen Rechenumgebungen jeweils eine Reihenfolge, anhand derer der Scheduler bestimmt, wo Jobs, die zur Ausführung bereit sind, ausgeführt werden sollen. Wenn die erste Rechenumgebung den Status hat `VALID` und über verfügbare Ressourcen verfügt, wird der Job für eine Container-Instance innerhalb dieser Rechenumgebung geplant. Wenn die erste Rechenumgebung den Status hat `INVALID` oder keine geeignete Rechenressource bereitstellen kann, versucht der Scheduler, den Job in der nächsten Rechenumgebung auszuführen.

Themen

- Verwaltete Computerumgebungen
- Nicht verwaltete Computerumgebungen
- Erstellen Sie eine Rechenumgebung
- Ressource berechnen AMIs
- Verwenden Sie EC2 Amazon-Startvorlagen mit AWS Batch
- EC2 Konfigurationen
- Strategien zur Zuweisung von Instance-Typen für AWS Batch
- Speicherverwaltung für Rechenressourcen
- Aktualisierung von Rechenumgebungen
- Fargate-Computerumgebungen
- Amazon EKS-Rechenumgebungen

Verwaltete Computerumgebungen

Sie können eine verwaltete Rechenumgebung verwenden, um die Kapazität und die Instanztypen der Rechenressourcen innerhalb der Umgebung zu AWS Batch verwalten. Dies basiert auf den Spezifikationen für Rechenressourcen, die Sie bei der Erstellung der Rechenumgebung definieren. Sie können wählen, ob Sie Amazon EC2 On-Demand-Instances und Amazon EC2 Spot-Instances verwenden möchten. Oder Sie können alternativ die Kapazität von Fargate und Fargate Spot in Ihrer verwalteten Computerumgebung verwenden. Wenn Sie Spot-Instances verwenden, können Sie

optional einen Höchstpreis festlegen. Auf diese Weise werden Spot-Instances nur gestartet, wenn der Spot-Instance-Preis unter einem bestimmten Prozentsatz des On-Demand-Preises liegt.

> **Important**
>
> Fargate Spot-Instances werden auf Windows containers on AWS Fargate nicht unterstützt. Eine Job-Warteschlange wird blockiert, wenn ein FargateWindows Job an eine Job-Warteschlange weitergeleitet wird, die nur Fargate Spot-Computerumgebungen verwendet.

> **Important**
>
> AWS Batch erstellt und verwaltet mehrere AWS Ressourcen in Ihrem Namen und innerhalb Ihres Kontos, darunter Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets und Amazon ECS-Cluster. Diese verwalteten Ressourcen sind speziell konfiguriert, um einen optimalen AWS Batch Betrieb zu gewährleisten. Die manuelle Änderung dieser im Batch verwalteten Ressourcen kann, sofern nicht ausdrücklich in der AWS Batch Dokumentation angegeben, zu unerwartetem Verhalten führen, das zu einer `INVALID` Rechenumgebung, suboptimalem Instanzskalierungsverhalten, verzögerter Workload-Verarbeitung oder unerwarteten Kosten führt. Diese manuellen Änderungen können vom Service nicht deterministisch unterstützt werden. AWS Batch Verwenden Sie immer die unterstützte Batch APIs - oder Batch-Konsole, um Ihre Rechenumgebungen zu verwalten.

Verwaltete Rechenumgebungen starten EC2 Amazon-Instances in der VPC und den Subnetzen, die Sie angeben, und registrieren sie dann bei einem Amazon ECS-Cluster. Die EC2 Amazon-Instances benötigen externen Netzwerkzugriff, um mit dem Amazon ECS-Serviceendpunkt zu kommunizieren. Einige Subnetze stellen EC2 Amazon-Instances keine öffentlichen IP-Adressen zur Verfügung. Wenn Ihre EC2 Amazon-Instances keine öffentliche IP-Adresse haben, müssen sie Network Address Translation (NAT) verwenden, um diesen Zugriff zu erhalten. Weitere Informationen finden Sie unter NAT-Gateways im Amazon VPC-Benutzerhandbuch. Weitere Informationen zum Erstellen einer VPC finden Sie unterErstellen einer Virtual Private Cloud .

Standardmäßig verwenden AWS Batch verwaltete Computerumgebungen eine aktuelle, genehmigte Version des für Amazon ECS optimierten AMI für Rechenressourcen. Möglicherweise möchten Sie jedoch aus verschiedenen Gründen Ihr eigenes AMI erstellen, das Sie für Ihre verwalteten

Computerumgebungen verwenden können. Weitere Informationen finden Sie unter Ressource berechnen AMIs.

> **Note**
>
> AWS Batch aktualisiert das AMIs in einer Rechenumgebung nicht automatisch, nachdem es erstellt wurde. Beispielsweise aktualisiert es die AMIs in Ihrer Computerumgebung nicht, wenn eine neuere Version des für Amazon ECS optimierten AMI veröffentlicht wird. Sie sind für die Verwaltung des Gastbetriebssystems verantwortlich. Dazu gehören alle Updates und Sicherheitspatches. Sie sind auch für jede zusätzliche Anwendungssoftware oder Hilfsprogramme verantwortlich, die Sie auf den Rechenressourcen installieren. Es gibt zwei Möglichkeiten, ein neues AMI für Ihre AWS Batch Jobs zu verwenden. Die ursprüngliche Methode besteht darin, die folgenden Schritte auszuführen:
>
> 1. Erstellen Sie eine neue Datenverarbeitungsumgebung mit dem neuen AMI.
> 2. Fügen Sie die Datenverarbeitungsumgebung einer vorhandenen Auftragswarteschlange hinzu.
> 3. Entfernen Sie die alte Datenverarbeitungsumgebung aus Ihrer Auftragswarteschlange.
> 4. Löschen Sie die alte Datenverarbeitungsumgebung.
>
> Im April 2022 AWS Batch wurde erweiterte Unterstützung für die Aktualisierung von Computerumgebungen hinzugefügt. Weitere Informationen finden Sie unter Aktualisierung von Rechenumgebungen. Um die erweiterte Aktualisierung von Rechenumgebungen für Updates zu nutzen AMIs, folgen Sie diesen Regeln:
>
> - Legen Sie den Parameter service role (`serviceRole`) entweder nicht fest oder legen Sie ihn auf die AWSServiceRoleForBatchdienstverknüpfte Rolle fest.
> - Setzen Sie den Parameter Allocation Strategy (`allocationStrategy`) auf`BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` oder`SPOT_PRICE_CAPACITY_OPTIMIZED`.
> - Stellen Sie den Parameter Update auf die neueste Image-Version (`updateToLatestImageVersion`) auf ein`true`.
> - Geben Sie keine AMI-ID in `imageId`, `imageIdOverride`(in `ec2Configuration`) oder in der Startvorlage (`launchTemplate`) an. AWS Batch Wählt in diesem Fall das neueste für Amazon ECS optimierte AMI aus, das AWS Batch zum Zeitpunkt der Initiierung des Infrastruktur-Updates unterstützt wird. Alternativ können Sie die AMI-ID in den

`imageIdOverride` Parametern `imageId` oder oder die durch die `LaunchTemplate` Eigenschaften identifizierte Startvorlage angeben. Wenn Sie eine dieser Eigenschaften ändern, wird ein Infrastruktur-Update gestartet. Wenn die AMI-ID in der Startvorlage angegeben ist, kann sie nicht durch die Angabe einer AMI-ID in den `imageIdOverride` Parametern `imageId` oder ersetzt werden. Sie kann nur durch Angabe einer anderen Startvorlage ersetzt werden. Oder, wenn die Version der Startvorlage auf `$Default` oder festgelegt ist`$Latest`, indem Sie entweder eine neue Standardversion für die Startvorlage festlegen (falls vorhanden`$Default`) oder indem Sie der Startvorlage eine neue Version hinzufügen (falls vorhanden`$Latest`).

Wenn diese Regeln befolgt werden, führt jedes Update, das ein Infrastruktur-Update startet, dazu, dass die AMI-ID erneut ausgewählt wird. Wenn die `version`Einstellung in der Startvorlage (`launchTemplate`) auf `$Latest` oder gesetzt ist`$Default`, wird die neueste Version oder Standardversion der Startvorlage zum Zeitpunkt des Infrastruktur-Updates ausgewertet, auch wenn sie nicht aktualisiert `launchTemplate`wurde.

Überlegungen bei der Erstellung parallel Jobs mit mehreren Knoten

AWS Batch empfiehlt, dedizierte Rechenumgebungen für die Ausführung von Multi-Node-Paralleljobs (MNP) und Nicht-MNP-Jobs zu erstellen. Dies liegt an der Art und Weise, wie Rechenkapazität in Ihrer verwalteten Computerumgebung geschaffen wird. Wenn Sie beim Erstellen einer neuen verwalteten Rechenumgebung einen `minvCpu` Wert größer als Null angeben, AWS Batch wird ein Instanzpool erstellt, der nur für Nicht-MNP-Jobs verwendet werden kann. Wenn ein parallel Auftrag mit mehreren Knoten eingereicht wird, wird neue Instanzkapazität für die Ausführung der parallel Jobs mit mehreren Knoten AWS Batch erstellt. In Fällen, in denen sowohl Einzelknoten- als auch Paralleljobs mit mehreren Knoten in derselben Rechenumgebung ausgeführt werden, in der entweder ein `minvCpus` oder `maxvCpus` festgelegt ist, AWS Batch wird, falls die erforderlichen Rechenressourcen nicht verfügbar sind, bis die aktuellen Jobs abgeschlossen sind, bevor die Rechenressourcen erstellt werden, die für die Ausführung der neuen Jobs erforderlich sind.

Nicht verwaltete Computerumgebungen

In einer nicht verwalteten Datenverarbeitungsumgebung verwalten Sie Ihre eigenen Datenverarbeitungsressourcen. Sie müssen überprüfen, ob das AMI, das Sie für Ihre Rechenressourcen verwenden, der AMI-Spezifikation für Amazon ECS-Container-Instances

entspricht. Weitere Informationen erhalten Sie unter AMI-Spezifikation für Rechenressourcen und Tutorial: Erstellen Sie ein Compute-Ressourcen-AMI.

> **Note**
>
> AWS Fargate-Ressourcen werden in nicht verwalteten Computerumgebungen nicht unterstützt.

Nachdem Sie Ihre nicht verwaltete Rechenumgebung erstellt haben, verwenden Sie den DescribeComputeEnvironmentsAPI-Vorgang, um die Details der Rechenumgebung anzuzeigen. Suchen Sie den Amazon ECS-Cluster, der mit der Umgebung verknüpft ist, und starten Sie dann Ihre Container-Instances manuell in diesem Amazon ECS-Cluster.

Der folgende AWS CLI Befehl stellt auch den Amazon ECS-Cluster-ARN bereit.

```
$ aws batch describe-compute-environments \
    --compute-environments unmanagedCE \
    --query "computeEnvironments[].ecsClusterArn"
```

Weitere Informationen finden Sie unter Starten einer Amazon ECS-Container-Instance im Amazon Elastic Container Service-Entwicklerhandbuch. Wenn Sie Ihre Rechenressourcen starten, geben Sie den Amazon ECS-Cluster-ARN an, den die Ressourcen mit den folgenden EC2 Amazon-Benutzerdaten registrieren. *ecsClusterArn*Ersetzen Sie durch den Cluster-ARN, den Sie mit dem vorherigen Befehl erhalten haben.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

Erstellen Sie eine Rechenumgebung

Bevor Sie Jobs ausführen können AWS Batch, müssen Sie eine Rechenumgebung erstellen. Sie können eine verwaltete Rechenumgebung erstellen, in der die EC2 Amazon-Instances oder AWS Fargate-Ressourcen innerhalb der Umgebung auf der Grundlage Ihrer Spezifikationen AWS Batch verwaltet werden. Alternativ können Sie auch eine nicht verwaltete Rechenumgebung erstellen, in der Sie die EC2 Amazon-Instance-Konfiguration innerhalb der Umgebung verwalten.

> **Important**
>
> Fargate Spot-Instances werden in den folgenden Szenarien nicht unterstützt:
>
> - Auf Amazon Linux-Containern mit ARM64 Architektur.
> - Windows containers on AWS Fargate
>
> In diesen Szenarien wird eine Job-Warteschlange blockiert, wenn ein Job an eine Job-Warteschlange weitergeleitet wird, die nur Fargate Spot-Computerumgebungen verwendet.

Themen

- Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mit Fargate-Ressourcen
- Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von EC2 Amazon-Ressourcen
- Tutorial: Erstellen Sie eine nicht verwaltete Rechenumgebung mithilfe von Amazon-Ressourcen EC2
- Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von Amazon EKS-Ressourcen
- Ressource: Vorlage für die Rechenumgebung

Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mit Fargate-Ressourcen

Gehen Sie wie folgt vor, um mithilfe von AWS Fargate Ressourcen eine verwaltete Rechenumgebung zu erstellen.

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Datenverarbeitungs-Umgebungen aus.
4. Wählen Sie Erstellen aus.
5. Konfigurieren Sie die Rechenumgebung.

> **Note**
>
> Rechenumgebungen für Windows containers on AWS Fargate Jobs müssen mindestens eine vCPU haben.

a. Wählen Sie für die Konfiguration der Rechenumgebung Fargate aus.

b. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

c. Wählen Sie unter Servicerolle die dienstbezogene Rolle aus, mit der der AWS Batch Dienst die erforderlichen AWS API-Operationen in Ihrem Namen aufrufen kann. Wählen Sie zum Beispiel aus AWSServiceRoleForBatch. Weitere Informationen finden Sie unter Dienstbezogene Rollenberechtigungen für AWS Batch.

d. (Optional) Erweitern Sie Tags. Um einen Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen). Geben Sie dann einen Schlüsselnamen und optional einen Wert ein. Wählen Sie Add tag.

e. Wählen Sie „Nächste Seite“.

6. Gehen Sie im Abschnitt Instanzkonfiguration wie folgt vor:

   a. (Optional) Aktivieren Sie Fargate Spot, um Fargate Spot-Kapazität verwenden zu können. Informationen zu Fargate Spot finden Sie unter Amazon Spot verwenden und EC2 Fargate_SPOT.

   b. Wählen Sie für Maximum v die maximale Anzahl von v ausCPUs, auf CPUs die Ihre Rechenumgebung skaliert werden kann, unabhängig von der Nachfrage in der Jobwarteschlange.

   c. Wählen Sie „Nächste Seite“.

7. Konfigurieren Sie das Netzwerk.

> **Important**
>
> Datenverarbeitungsressourcen benötigen einen externen Netzwerkzugriff, um mit dem Amazon-ECS-Service-Endpunkt zu kommunizieren. Dies kann über einen Schnittstellen-VPC-Endpunkt oder über Ihre Datenverarbeitungsressourcen mit öffentlichen IP-Adressen sein.

Weitere Informationen zu Interface-VPC-Endpunkten finden Sie unter Amazon-ECS-Schnittstellen-VPC-Endpunkte (AWS PrivateLink) im Amazon Elastic Container Service-Entwicklerhandbuch.
Wenn Sie keinen Schnittstellen-VPC-Endpunkt konfiguriert haben und Ihre Datenverarbeitungsressourcen keine öffentlichen IP-Adressen haben, müssen Sie Netzwerkadressübersetzung (Network Address Translation, NAT) verwenden, um diesen Zugriff zur Verfügung zu stellen. Weitere Informationen finden Sie unter NAT-Gateways im Amazon VPC-Benutzerhandbuch. Weitere Informationen finden Sie unter the section called “Erstellen einer VPC”.

a. Wählen Sie für Virtual Private Cloud (VPC) ID eine VPC aus, in der Sie Ihre Instances starten möchten.

b. Wählen Sie für Subnetze die zu verwendenden Subnetze aus. Standardmäßig sind alle Subnetze innerhalb der ausgewählten VPC verfügbar.

> **Note**
>
> AWS Batch on Fargate unterstützt derzeit keine Local Zones. Weitere Informationen finden Sie unter Amazon ECS-Cluster in Local Zones, Wavelength Zones und AWS Outposts im Amazon Elastic Container Service Developer Guide.

c. Wählen Sie für Sicherheitsgruppen eine Sicherheitsgruppe aus, um sie Ihren Instances anzufügen. Standardmäßig ist die Standardsicherheitsgruppe für Ihre VPC ausgewählt.

d. Wählen Sie „Nächste Seite“.

8. Überprüfen Sie die Konfigurationsschritte zur Überprüfung. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Create Compute Environment aus.

Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von EC2 Amazon-Ressourcen

Gehen Sie wie folgt vor, um eine verwaltete Rechenumgebung mithilfe von Amazon Elastic Compute Cloud (Amazon EC2) -Ressourcen zu erstellen.

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.

2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Datenverarbeitungs-Umgebungen aus.
4. Wählen Sie Erstellen aus.
5. Konfigurieren Sie die Umgebung.
   a. Wählen Sie für die Konfiguration der Rechenumgebung Amazon Elastic Compute Cloud (Amazon EC2).
   b. Wählen Sie als Orchestrierungstyp die Option Verwaltet aus.
   c. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
   d. Wählen Sie für die Servicerolle die serviceverknüpfte Rolle aus, mit der der AWS Batch Dienst in Ihrem Namen die erforderlichen AWS API-Operationen aufrufen kann. Wählen Sie zum Beispiel aus AWSServiceRoleForBatch. Weitere Informationen finden Sie unter Dienstbezogene Rollenberechtigungen für AWS Batch.
   e. Wählen Sie im Feld Instance-Role (Instance-Rolle) ein neues Instance-Profil aus oder verwenden Sie ein bestehendes Instance-Profil, bei dem die erforderlichen IAM-Berechtigungen angefügt sind. Dieses Instance-Profil ermöglicht es den Amazon ECS-Container-Instances, die für Ihre Rechenumgebung erstellt wurden, die erforderlichen AWS API-Operationen in Ihrem Namen aufzurufen. Weitere Informationen finden Sie unter Amazon ECS-Instance-Rolle. Wenn Sie ein neues Instance-Profil erstellen, wird die erforderliche Rolle (`ecsInstanceRole`) für Sie erstellt.
   f. (Optional) Erweitern Sie Tags.
   g. (Optional) Wählen Sie für EC2 Tags die Option Tag hinzufügen aus, um Ressourcen, die in der Rechenumgebung gestartet werden, ein Tag hinzuzufügen. Geben Sie dann einen Schlüsselnamen und optional einen Wert ein. Wählen Sie Add tag.
   h. (Optional) Wählen Sie für Tags die Option Tag hinzufügen aus. Geben Sie dann einen Schlüsselnamen und optional einen Wert ein. Wählen Sie Add tag.

      Weitere Informationen finden Sie unter Kennzeichnen Sie Ihre AWS Batch Ressourcen.
   i. Wählen Sie „Nächste Seite“.
6. Gehen Sie im Abschnitt Instanzkonfiguration wie folgt vor:
   a. (Optional) Aktivieren Sie unter Verwendung von Spot-Instances die Option Spot. Weitere Informationen finden Sie unter Spot-Instances.

b. (Nur Spot) Wählen Sie für den On-Demand-Preis von maximal% den maximalen Prozentsatz aus, den ein Spot-Instance-Preis im Vergleich zum On-Demand-Preis für diesen Instance-Typ vor dem Start der Instances betragen kann. Wenn Ihr Höchstpreis beispielsweise 20% beträgt, muss der Spot-Preis weniger als 20% des aktuellen On-Demand-Preises für diese EC2 Instance betragen. Sie zahlen immer nur den niedrigsten (Markt-) Preis und niemals mehr als Ihren maximalen Prozentsatz. Wenn Sie dieses Feld leer lassen, ist der Standardwert 100 % des On-Demand-Preises.

c. (Nur Spot) Wählen Sie für die Spot-Flottenrolle eine bestehende Amazon EC2 Spot Fleet IAM-Rolle aus, die Sie auf Ihre Spot-Computerumgebung anwenden möchten. Wenn Sie noch keine bestehende Amazon EC2 Spot Fleet IAM-Rolle haben, müssen Sie zuerst eine erstellen. Weitere Informationen finden Sie unter Rolle bei Amazon EC2 Spot Fleet.

> **Important**
>
> Um Ihre Spot-Instances bei der Erstellung zu taggen, muss Ihre Amazon EC2 Spot Fleet IAM-Rolle die neuere von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie verwenden. Die von Amazon EC2 SpotFleetRole verwaltete Richtlinie verfügt nicht über die erforderlichen Berechtigungen, um Spot-Instances zu taggen. Weitere Informationen erhalten Sie unter Spot-Instances wurden bei der Erstellung nicht markiert und the section called “Markieren Ihrer -Ressourcen mit Tags (Markierungen)”.

d. Wählen Sie für Minimum v CPUs die Mindestanzahl von v ausCPUs , die Ihre Rechenumgebung unabhängig von der Nachfrage in der Job-Warteschlange beibehält.

e. Wählen Sie für CPUsDesired v die Anzahl von v ausCPUs , mit der Ihre Rechenumgebung gestartet wird. Wenn die Nachfrage in Ihrer Job-Warteschlange steigt, AWS Batch können Sie die gewünschte Anzahl von v CPUs in Ihrer Rechenumgebung erhöhen und EC2 Instanzen hinzufügen, bis die maximale Anzahl v erreicht istCPUs. Bei sinkender Nachfrage AWS Batch kann die gewünschte Anzahl von v CPUs in Ihrer Rechenumgebung verringert und Instanzen bis auf das Minimum v entfernt CPUs werden.

f. Wählen Sie für Maximum v die maximale Anzahl von v ausCPUs, auf CPUs die Ihre Rechenumgebung skaliert werden kann, unabhängig von der Nachfrage in der Jobwarteschlange.

g. Wählen Sie unter Zulässige Instance-Typen die EC2 Amazon-Instance-Typen aus, die gestartet werden können. Sie können Instance-Familien angeben, um jeden beliebigen Instance-Typ innerhalb dieser Familien zu starten (z. B. `c5c5n`,, oder`p3`). Sie können auch

bestimmte Größen innerhalb einer Familie angeben (z. B.`c5.8xlarge`). Instanztypen aus Metall gehören nicht zu den Instanzfamilien. Beinhaltet zum Beispiel `c5` nicht`c5.metal`. Sie können auch Instance-Typen (aus den Instance-Familien C4M4, undR4) auswählen, die den Anforderungen Ihrer Job-Warteschlangen entsprechen. `optimal`

> **Note**
>
> Wenn Sie eine Compute-Umgebung erstellen, müssen die Instance-Typen, die Sie für die Compute-Umgebung auswählen, dieselbe Architektur verwenden. Beispielsweise ist es nicht möglich, x86- und ARM-Instances in derselben Compute-Umgebung zu kombinieren.

> **Note**
>
> AWS Batch wird auf der GPUs Grundlage der erforderlichen Anzahl in Ihren Job-Warteschlangen skaliert. Um die GPU-Planung verwenden zu können, muss die Rechenumgebung Instance-Typen aus den `g6` Familien `p3` `p4p5`,`p6`,`g3`,`g3s`,`g4`,`g5`, oder enthalten.

> **Note**
>
> `optimal`Verwendet derzeit Instanztypen aus den R4 Instanzfamilien C4M4, und. Falls AWS-Regionen es keine Instanztypen aus diesen Instanzfamilien gibt, werden Instanztypen aus den R5 Instanzfamilien C5M5, und verwendet.

h. Erweitern Sie Additional configuration (Zusätzliche Konfiguration).

i. (Optional) Geben Sie für Placement-Gruppe einen Namen für die Placement-Gruppe ein, um Ressourcen in der Rechenumgebung zu gruppieren.

j. (Optional) Wählen Sie für das EC2 key pair ein öffentliches und ein privates key pair als Sicherheitsanmeldedaten aus, wenn Sie eine Verbindung mit der Instance herstellen. Weitere Informationen zu EC2 Amazon-Schlüsselpaaren finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

k. Wählen Sie als Allocation strategy (Zuordnungsstrategie) die Zuordnungsstrategie aus, die bei der Auswahl von Instance-Typen aus der Liste der zulässigen Instance-Typen

verwendet werden soll. BEST_FIT_PROGRESSIVE ist normalerweise die bessere Wahl für EC2 On-Demand-Rechenumgebungen, SPOT_CAPACITY_OPTIMIZED und SPOT_PRICE_CAPACITY_OPTIMIZED für Spot-Rechenumgebungen. EC2 Weitere Informationen finden Sie unter the section called “Strategien für die Zuweisung von Instance-Typen”.

l. (Optional) Wählen Sie für die EC2 Konfiguration die Werte Image-Typ und Image-ID Override aus, AWS Batch um Informationen zur Auswahl von Amazon Machine Images (AMIs) für Instances in der Rechenumgebung bereitzustellen. Wenn die Image-ID-Override nicht für jeden Image-Typ angegeben ist, AWS Batch wird ein aktuelles Amazon ECS-optimiertes AMI ausgewählt. Wenn kein Image-Typ angegeben ist, ist der Standard Amazon Linux 2 für Instances ohne GPU oder AWS Graviton.

> **Important**
>
> Um ein benutzerdefiniertes AMI zu verwenden, wählen Sie den Image-Typ aus und geben Sie dann die benutzerdefinierte AMI-ID in das Feld Image-ID Override ein.

Amazon Linux 2

Standard für alle AWS Graviton-basierten Instance-Familien (z. B., `C6g` `M6gR6g`, und`T4g`) und kann für alle Instance-Typen ohne GPU verwendet werden.

Amazon Linux 2 (GPU)

Standard für alle GPU-Instanzfamilien (zum Beispiel `P4` und`G4`) und kann für alle Instance-Typen verwendet werden, die nicht auf AWS Graviton basieren.

Amazon Linux

Kann für Instanzfamilien ohne GPU oder AWS Graviton verwendet werden. Die Standardunterstützung für Amazon Linux AMI wurde eingestellt. Weitere Informationen finden Sie unter Amazon Linux AMI.

> **Note**
>
> Das AMI, das Sie für eine Compute-Umgebung auswählen, muss der Architektur der Instance-Typen entsprechen, die Sie für die betreffende Compute-Umgebung verwenden möchten. Wenn Ihre Datenverarbeitungsumgebung beispielsweise

A1-Instance-Typen verwendet, muss das von Ihnen ausgewählte Compute-Ressourcen-AMI ARM-Ressourcen unterstützen. Amazon ECS verkauft sowohl x86- als auch ARM-Versionen des für Amazon ECS optimierten Amazon Linux 2-AMI. Weitere Informationen finden Sie unter Amazon ECS-optimiertes Amazon Linux 2-AMI im Amazon Elastic Container Service Developer Guide.

m. (Optional) Wählen Sie unter Startvorlage eine bestehende EC2 Amazon-Startvorlage aus, um Ihre Rechenressourcen zu konfigurieren. Die Standardversion der Vorlage wird automatisch ausgefüllt. Weitere Informationen finden Sie unter Verwenden Sie EC2 Amazon-Startvorlagen mit AWS Batch.

> **Note**
>
> In einer Startvorlage können Sie ein benutzerdefiniertes AMI angeben, das Sie erstellt haben.

n. (Optional) Geben Sie unter Launch template version (Version der Startvorlage) `$Default`, `$Latest` oder eine bestimmte Versionsnummer ein, die verwendet werden soll.

> **Note**
>
> Hinweis: Wenn Sie eine der Ersatzvariablen ($Default oder $Latest) verwenden, wird die aktuelle Standard- oder neueste Versionsnummer zum Zeitpunkt des Speicherns dieser Konfiguration verwendet. Wenn sich die Standard- oder neueste Version in future ändert, müssen Sie die Informationen aktualisieren — sie werden nicht automatisch aktualisiert.

> **Important**
>
> Wenn der Versionsparameter der Startvorlage `$Default` oder ist`$Latest`, wird die Standard- oder neueste Version der angegebenen Startvorlage bei einem Infrastruktur-Update bewertet. Wenn standardmäßig eine andere AMI-ID ausgewählt ist oder die neueste Version der Startvorlage ausgewählt ist, wird diese AMI-ID im Update verwendet. Weitere Informationen finden Sie unter the section called “Aktualisierung der AMI-ID”.

o. (Optional) Wählen Sie für Override Launch Templates eine bestehende EC2 Amazon-Startvorlage aus, die für bestimmte Instance-Typen und Familien verwendet werden soll.

p. (Optional) Geben Sie für Override Launch Template-Version eine bestimmte Versionsnummer ein, die verwendet werden soll`$Default`, oder`$Latest`.

> **Note**
>
> Wenn Sie entweder die `$Latest` Variable `$Default` oder verwenden, AWS Batch werden die aktuellen Informationen zum Zeitpunkt der Erstellung der Rechenumgebung übernommen. Wenn sich die Standard- oder neueste Version in future ändert, müssen Sie die Informationen über UpdateComputeEnvironmentoder über das AWS Management Console - aktualisieren AWS Batch.

q. (Optional) Wählen Sie für Target-Instance-Typen den Instance-Typ oder die Instance-Familie aus, auf die Sie die Override-Startvorlage anwenden möchten.

> **Note**
>
> Wenn Sie eine Startvorlage zum Außerkraftsetzen angeben, ist die Angabe der Target-Instanztypen erforderlich. Weitere Informationen finden Sie unter LaunchTemplateSpecificationOverride. targetInstanceTypes.

> **Note**
>
> Wenn der Instanztyp oder die Familie, die Sie auswählen möchten, nicht in dieser Liste angezeigt wird, überprüfen Sie die Auswahl, die Sie in `Allowed instance types` getroffen haben.

r. Wählen Sie „Nächste Seite“.

7. Gehen Sie im Abschnitt Netzwerkkonfiguration wie folgt vor:

> **Important**
>
> Datenverarbeitungsressourcen benötigen einen externen Netzwerkzugriff, um mit dem Amazon-ECS-Service-Endpunkt zu kommunizieren. Dies kann über einen Schnittstellen-

VPC-Endpunkt oder über Ihre Datenverarbeitungsressourcen mit öffentlichen IP-Adressen sein.
Weitere Informationen zu Interface-VPC-Endpunkten finden Sie unter Amazon-ECS-Schnittstellen-VPC-Endpunkte (AWS PrivateLink) im Amazon Elastic Container Service-Entwicklerhandbuch.
Wenn Sie keinen Schnittstellen-VPC-Endpunkt konfiguriert haben und Ihre Datenverarbeitungsressourcen keine öffentlichen IP-Adressen haben, müssen Sie Netzwerkadressübersetzung (Network Address Translation, NAT) verwenden, um diesen Zugriff zur Verfügung zu stellen. Weitere Informationen finden Sie unter NAT-Gateways im Amazon VPC-Benutzerhandbuch. Weitere Informationen finden Sie unter the section called “Erstellen einer VPC”.

a. Wählen Sie für Virtual Private Cloud (VPC) ID eine VPC aus, auf der Ihre Instances gestartet werden sollen.

b. Wählen Sie für Subnetze die zu verwendenden Subnetze aus. Standardmäßig sind alle Subnetze innerhalb der ausgewählten VPC verfügbar.

> **Note**
>
> AWS Batch bei Amazon EC2 unterstützt Local Zones. Weitere Informationen finden Sie unter Local Zones im Amazon EC2 User Guide und Amazon ECS-Cluster in Local Zones, Wavelength Zones und AWS Outposts im Amazon Elastic Container Service Developer Guide.

c. (Optional) Wählen Sie für Sicherheitsgruppen eine Sicherheitsgruppe aus, die Sie Ihren Instances zuordnen möchten. Standardmäßig ist die Standardsicherheitsgruppe für Ihre VPC ausgewählt.

> **Note**
>
> Hinweis: Wenn Sie eine der Ersatzvariablen ($Default oder $Latest) verwenden, wird die aktuelle Standard- oder neueste Versionsnummer zum Zeitpunkt des Speicherns dieser Konfiguration verwendet. Wenn sich die Standard- oder neueste Version in future ändert, müssen Sie die Informationen aktualisieren — sie werden nicht automatisch aktualisiert.

8. Wählen Sie „Nächste Seite“.
9. Überprüfen Sie die Konfigurationsschritte zur Überprüfung. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Create Compute Environment aus.

Tutorial: Erstellen Sie eine nicht verwaltete Rechenumgebung mithilfe von Amazon-Ressourcen EC2

Gehen Sie wie folgt vor, um mithilfe von Amazon Elastic Compute Cloud (Amazon EC2) -Ressourcen eine nicht verwaltete Rechenumgebung zu erstellen.

1. Öffnen Sie die AWS Batch Konsole unter. https://console.aws.amazon.com/batch/
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie auf der Seite Compute Environments die Option Create aus.
4. Konfigurieren Sie die Umgebung.
   a. Wählen Sie für die Konfiguration der Rechenumgebung Amazon Elastic Compute Cloud (Amazon EC2).
   b. Wählen Sie als Orchestrierungstyp die Option Unmanaged aus.
5. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
6. Wählen Sie unter Servicerolle eine Rolle aus, mit der der AWS Batch Dienst in Ihrem Namen die erforderlichen AWS API-Operationen aufrufen kann.

   > **Note**
   > Sie können es nicht `AWSServiceRoleForBatch` für nicht verwaltete Rechenumgebungen verwenden.

7. Wählen Sie für Maximum v die maximale Anzahl von v ausCPUs, auf CPUs die Ihre Rechenumgebung skaliert werden kann, unabhängig von der Nachfrage in der Jobwarteschlange.

8. (Optional) Erweitern Sie Tags. Um einen Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen). Geben Sie dann einen Schlüsselnamen und optional einen Wert ein. Wählen Sie Add tag. Weitere Informationen finden Sie unter Kennzeichnen Sie Ihre AWS Batch Ressourcen.
9. Wählen Sie „Nächste Seite“.
10. Überprüfen Sie die Konfigurationsschritte zur Überprüfung. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Create Compute Environment aus.

Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von Amazon EKS-Ressourcen

Gehen Sie wie folgt vor, um eine verwaltete Rechenumgebung mithilfe von Amazon Elastic Kubernetes Service (Amazon EKS) -Ressourcen zu erstellen.

1. Öffnen Sie die AWS Batch Konsole unter. https://console.aws.amazon.com/batch/
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Datenverarbeitungs-Umgebungen aus.
4. Wählen Sie Erstellen aus.
5. Wählen Sie für die Konfiguration der Rechenumgebung Amazon Elastic Kubernetes Service (Amazon EKS).
6. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung an. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
7. Wählen Sie unter Instanzrolle ein vorhandenes Instanzprofil aus, dem die erforderlichen IAM-Berechtigungen zugeordnet sind.

   > **Note**
   >
   > Um eine Rechenumgebung in der AWS Batch Konsole zu erstellen, wählen Sie ein Instanzprofil mit den `eks:DescribeCluster` Berechtigungen `eks:ListClusters` und aus.

8. Wählen Sie für EKS-Cluster einen vorhandenen Amazon EKS-Cluster aus.
9. Geben Sie für Namespace einen Kubernetes Namespace ein, um Ihre AWS Batch Prozesse im Cluster zu gruppieren.

10. (Optional) Erweitern Sie Tags. Wählen Sie Tag hinzufügen und geben Sie dann ein Schlüssel-Wert-Paar ein.
11. Wählen Sie „Nächste Seite“.
12. (Optional) Aktivieren Sie für EC2 Spot-Instances verwenden die Option Verwendung von Spot-Instances aktivieren, um Amazon EC2 Spot-Instances zu verwenden.
13. (Nur Spot) Wählen Sie für den On-Demand-Preis von maximal% den maximalen Prozentsatz aus, den ein Spot-Instance-Preis im Vergleich zum On-Demand-Preis für diesen Instance-Typ vor dem Start der Instances betragen kann. Wenn Ihr Höchstpreis beispielsweise 20% beträgt, muss der Spot-Preis weniger als 20% des aktuellen On-Demand-Preises für diese EC2 Instance betragen. Sie zahlen immer nur den niedrigsten (Markt-) Preis und niemals mehr als Ihren maximalen Prozentsatz. Wenn Sie dieses Feld leer lassen, ist der Standardwert 100 % des On-Demand-Preises.
14. (Nur Spot) Wählen Sie für die Spot-Flottenrolle die Amazon EC2 Spot-Flotten-IAM-Rolle für die `SPOT` Rechenumgebung aus.

    > **Important**
    >
    > Diese Rolle ist erforderlich, wenn die Zuweisungsstrategie auf eingestellt `BEST_FIT` oder nicht angegeben ist.

15. (Optional) Wählen Sie für Minimum v die Mindestanzahl von v ausCPUs, CPUs die Ihre Rechenumgebung unabhängig von der Nachfrage in der Auftragswarteschlange verwaltet.
16. (Optional) Wählen Sie für Maximum v die maximale Anzahl von v ausCPUs, auf CPUs die Ihre Rechenumgebung skaliert werden kann, unabhängig von der Nachfrage in der Auftragswarteschlange.
17. Wählen Sie unter Zulässige Instance-Typen die EC2 Amazon-Instance-Typen aus, die gestartet werden können. Sie können Instance-Familien angeben, um jeden beliebigen Instance-Typ innerhalb dieser Familien zu starten (z. B. `c5c5n`,, oder`p3`). Sie können auch bestimmte Größen innerhalb einer Familie angeben (z. B.`c5.8xlarge`). Instanztypen aus Metall gehören nicht zu den Instanzfamilien. Beinhaltet zum Beispiel `c5` nicht`c5.metal`. Sie können auch Instance-Typen (aus den Instance-Familien C4M4, und`R4`) auswählen, da Sie diese benötigen, die den Anforderungen Ihrer Job-Warteschlangen entsprechen. `optimal`

> **Note**
>
> Wenn Sie eine Compute-Umgebung erstellen, müssen die Instance-Typen, die Sie für die Compute-Umgebung auswählen, dieselbe Architektur verwenden. Beispielsweise ist es nicht möglich, x86- und ARM-Instances in derselben Compute-Umgebung zu kombinieren.

> **Note**
>
> AWS Batch skaliert auf der GPUs Grundlage der erforderlichen Anzahl in Ihren Job-Warteschlangen. Um die GPU-Planung verwenden zu können, muss die Rechenumgebung Instance-Typen aus den `g6` Familien `p3` `p4p5`,`p6`,`g3`,`g3s`,`g4`,`g5`, oder enthalten.

> **Note**
>
> Derzeit nutzt `optimal` Instance-Typen aus den Instance-Familien C4, M4 und R4. Wenn AWS-Regionen es keine Instance-Typen aus diesen Instance-Familien gibt, werden Instance-Typen aus den R5 Instance-Familien C5M5, und verwendet.

18. (Optional) Erweitern Sie Zusätzliche Konfiguration.

    a. (Optional) Geben Sie unter Platzierungsgruppe einen Namen für die Platzierungsgruppe ein, um Ressourcen in der Rechenumgebung zu gruppieren.

    b. Wählen Sie als Zuweisungsstrategie BEST_FIT_PROGRESSIVE aus.

    c. (Optional) Wählen Sie für die Konfiguration von Amazon Machine Images (AMIs) die Option Amazon Machine Images (amis) -Konfiguration hinzufügen aus.

       Sie können entweder ein Amazon EKS-optimiertes Amazon Linux-AMI oder ein benutzerdefiniertes AMI verwenden.

       i. So verwenden Sie ein Amazon EKS-optimiertes Amazon Linux-AMI:

          A. Wählen Sie als Bildtyp eine der folgenden Optionen aus:

- Amazon Linux 2: Standard für alle AWS Graviton-basierten Instance-Familien (z. B., `C6g` `M6gR6g`, und`T4g`) und kann für alle Instance-Typen ohne GPU verwendet werden.
- Amazon Linux 2 (beschleunigt): Standard für alle GPU-Instance-Familien (z. B. `P4` und`G4`) und kann für alle Instance-Typen verwendet werden, die nicht auf AWS Graviton basieren.
- Amazon Linux 2023: AWS Batch unterstützt Amazon Linux 2023 (AL2023).
- Amazon Linux 2023 (beschleunigt): GPU-Instance-Familien, die für alle Instance-Typen verwendet werden können, die nicht AWS auf Graviton basieren.

B. Geben Sie für KubernetesVersion eine KubernetesVersionsnummer ein.

ii. Um ein benutzerdefiniertes AMI zu verwenden:

A. Wählen Sie unter Image-Typ den AMI-Typ aus, auf dem das benutzerdefinierte AMI basiert:

- Amazon Linux 2: Standard für alle AWS Graviton-basierten Instance-Familien (z. B., `C6g` `M6gR6g`, und`T4g`) und kann für alle Instance-Typen ohne GPU verwendet werden.
- Amazon Linux 2 (beschleunigt): Standard für alle GPU-Instance-Familien (z. B. `P4` und`G4`) und kann für alle Instance-Typen verwendet werden, die nicht auf AWS Graviton basieren.
- Amazon Linux 2023: AWS Batch unterstützt AL2 023.
- Amazon Linux 2023 (beschleunigt): GPU-Instance-Familien, die für alle Instance-Typen verwendet werden können, die nicht AWS auf Graviton basieren.

B. Geben Sie für Image ID Override die benutzerdefinierte AMI-ID ein.

C. Geben Sie für KubernetesVersion eine KubernetesVersionsnummer ein.

d. (Optional) Wählen Sie unter Startvorlage eine vorhandene Startvorlage aus.

e. (Optional) Geben Sie für Version der Startvorlage**`$Default`**,**`$Latest`**, oder eine Versionsnummer ein.

f. (Optional) Um eine Überschreibung hinzuzufügen, wählen Sie unter Startvorlage überschreiben die Option Startvorlage hinzufügen aus:

i. (Optional) Wählen Sie unter Startvorlage die Startvorlage aus, zu der Sie die Überschreibung hinzufügen möchten.

ii. (Optional) Wählen Sie für Launch Template Version die Versionsnummer der Startvorlage`$Default`, oder`$Latest`.

iii. (Optional) Wählen Sie für Target-Instance-Typen den Instance-Typ oder die Instance-Familie aus, auf die diese Überschreibung angewendet werden soll. Dies kann nur auf Instance-Typen und -Familien abzielen, die unter Zulässige Instance-Typen enthalten sind.

iv. (Optional) Wählen Sie für UserDataType die EKS-Knoteninitialisierung aus. Verwenden Sie dieses Feld nur, wenn Sie ein AMI entweder im Launch Template oder als Launch Template Override angegeben haben. Wählen Sie EKS_NODEADM für benutzerdefiniert AMIs basierend auf **`EKS_AL2023`** oder oder **`EKS_AL2023_NVIDIA`** oder EKS_BOOSTRAP_SH für und. `EKS_AL2` `EKS_AL_NVIDIA` Der Standardwert ist EKS_BOOSTRAP_SH.

    Sie würden UserDataType verwenden, wenn Sie eine gemischte Umgebung haben, in der Sie sowohl als auch AL2 benutzerdefinierte 023-basierte benutzerdefinierte Umgebungen in derselben Rechenumgebung verwenden. AL2 AMIs

19. Wählen Sie „Nächste Seite“.
20. Wählen Sie für Virtual Private Cloud (VPC) ID eine VPC aus, auf der die Instances gestartet werden sollen.
21. Wählen Sie für Subnetze die zu verwendenden Subnetze aus. Standardmäßig sind alle Subnetze innerhalb der ausgewählten VPC verfügbar.

    > **Note**
    >
    > AWS Batch auf Amazon EKS unterstützt Local Zones. Weitere Informationen finden Sie unter Amazon EKS and AWS Local Zones im Amazon EKS-Benutzerhandbuch.

22. (Optional) Wählen Sie für Sicherheitsgruppen eine Sicherheitsgruppe aus, die Sie Ihren Instances zuordnen möchten. Standardmäßig ist die Standardsicherheitsgruppe für Ihre VPC ausgewählt.
23. Wählen Sie Nächste Seite.
24. Überprüfen Sie die Konfigurationsschritte zur Überprüfung. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Create Compute Environment aus.

Ressource: Vorlage für die Rechenumgebung

Das folgende Beispiel zeigt eine leere Vorlage für eine Rechenumgebung. Sie können diese Vorlage verwenden, um Ihre Rechenumgebung zu erstellen, die dann in einer Datei gespeichert und mit der AWS CLI `--cli-input-json` Option verwendet werden kann. Weitere Informationen zu diesen Parametern finden Sie CreateComputeEnvironmentin der AWS Batch API-Referenz.

> **Note**
>
> Mit dem folgenden AWS CLI Befehl können Sie eine Vorlage für die Rechenumgebung generieren.
>
> ```
> $ aws batch create-compute-environment --generate-cli-skeleton
> ```

```
{
    "computeEnvironmentName": "",
    "type": "UNMANAGED",
    "state": "DISABLED",
    "unmanagedvCpus": 0,
    "computeResources": {
        "type": "EC2",
        "allocationStrategy": "BEST_FIT_PROGRESSIVE",
        "minvCpus": 0,
        "maxvCpus": 0,
        "desiredvCpus": 0,
        "instanceTypes": [
            ""
        ],
        "imageId": "",
        "subnets": [
            ""
        ],
        "securityGroupIds": [
            ""
        ],
        "ec2KeyPair": "",
        "instanceRole": "",
        "tags": {
            "KeyName": ""
        },
```

```
        "placementGroup": "",
        "bidPercentage": 0,
        "spotIamFleetRole": "",
        "launchTemplate": {
            "launchTemplateId": "",
            "launchTemplateName": "",
            "version": ""
        },
        "ec2Configuration": [
            {
                "imageType": "",
                "imageIdOverride": "",
                "imageKubernetesVersion": ""
            }
        ]
    },
    "serviceRole": "",
    "tags": {
        "KeyName": ""
    },
    "eksConfiguration": {
        "eksClusterArn": "",
        "kubernetesNamespace": ""
    }
}
```

Ressource berechnen AMIs

Standardmäßig verwenden AWS Batch verwaltete Computerumgebungen eine aktuelle, genehmigte Version des für Amazon ECS optimierten AMI für Rechenressourcen. Möglicherweise möchten Sie jedoch Ihr eigenes AMI erstellen, das Sie für Ihre verwalteten und nicht verwalteten Computerumgebungen verwenden können. Wenn Sie eines der folgenden Dinge benötigen, empfehlen wir Ihnen, Ihr eigenes AMI zu erstellen:

- Erhöhung der Speichergröße Ihrer AMI-Root- oder Datenvolumes
- Hinzufügen von Instance-Speichervolumes für unterstützte EC2 Amazon-Instance-Typen
- Anpassen des Amazon ECS-Container-Agenten
- Anpassen von Docker
- Konfiguration eines GPU-Workload-AMI, um Containern den Zugriff auf GPU-Hardware auf unterstützten EC2 Amazon-Instance-Typen zu ermöglichen

> **Note**
>
> Nachdem eine Datenverarbeitungsumgebung erstellt wurde, wird die AMIs in der Rechenumgebung AWS Batch nicht aktualisiert. AWS Batch aktualisiert die AMIs in Ihrer Computerumgebung auch nicht, wenn eine neuere Version des für Amazon ECS optimierten AMI verfügbar ist. Sie sind für die Verwaltung des Gastbetriebssystems verantwortlich. Dazu gehören alle Updates und Sicherheitspatches. Sie sind auch für jede zusätzliche Anwendungssoftware oder Hilfsprogramme verantwortlich, die Sie auf den Rechenressourcen installieren. Gehen Sie wie folgt vor, um ein neues AMI für Ihre AWS Batch Jobs zu verwenden:
>
> 1. Erstellen Sie eine neue Datenverarbeitungsumgebung mit dem neuen AMI.
> 2. Fügen Sie die Datenverarbeitungsumgebung einer vorhandenen Auftragswarteschlange hinzu.
> 3. Entfernen Sie die alte Datenverarbeitungsumgebung aus Ihrer Auftragswarteschlange.
> 4. Löschen Sie die alte Datenverarbeitungsumgebung.
>
> Im April 2022 AWS Batch wurde erweiterte Unterstützung für die Aktualisierung von Computerumgebungen hinzugefügt. Weitere Informationen finden Sie unter Aktualisierung von Rechenumgebungen. Um die erweiterte Aktualisierung von Rechenumgebungen für Updates zu nutzen AMIs, folgen Sie diesen Regeln:
>
> - Legen Sie den Parameter service role (`serviceRole`) entweder nicht fest oder legen Sie ihn auf die AWSServiceRoleForBatchdienstverknüpfte Rolle fest.
> - Setzen Sie den Parameter Allocation Strategy (`allocationStrategy`) auf `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, oder`SPOT_PRICE_CAPACITY_OPTIMIZED`.
> - Stellen Sie den Parameter Update auf die neueste Image-Version (`updateToLatestImageVersion`) auf ein`true`.
> - Geben Sie keine AMI-ID in `imageId`, `imageIdOverride`(in `ec2Configuration`) oder in der Startvorlage (`launchTemplate`) an. Wenn Sie keine AMI-ID angeben, AWS Batch wählt das neueste Amazon ECS-optimierte AMI aus, das zum Zeitpunkt der Initiierung des Infrastruktur-Updates AWS Batch unterstützt wird. Alternativ können Sie die AMI-ID in den `imageIdOverride` Parametern `imageId` oder angeben. Sie können auch die Startvorlage angeben, die anhand der `LaunchTemplate` Eigenschaften identifiziert wird. Wenn Sie eine dieser Eigenschaften ändern, wird ein Infrastruktur-Update gestartet. Wenn

die AMI-ID in der Startvorlage angegeben ist, kann die AMI-ID nicht durch die Angabe einer AMI-ID in den `imageIdOverride` Parametern `imageId` oder ersetzt werden. Die AMI-ID kann nur durch Angabe einer anderen Startvorlage ersetzt werden. Wenn die Version der Startvorlage auf `$Default` oder gesetzt ist`$Latest`, kann die AMI-ID ersetzt werden, indem entweder eine neue Standardversion für die Startvorlage festgelegt wird (if`$Default`) oder indem der Startvorlage eine neue Version hinzugefügt wird (if`$Latest`).

Wenn diese Regeln befolgt werden, führt jedes Update, das ein Infrastruktur-Update startet, dazu, dass die AMI-ID erneut ausgewählt wird. Wenn die `version`Einstellung in der Startvorlage (`launchTemplate`) auf `$Latest` oder gesetzt ist`$Default`, wird die neueste oder Standardversion der Startvorlage zum Zeitpunkt des Infrastruktur-Updates ausgewertet, auch wenn sie `launchTemplate`nicht aktualisiert wurde.

Themen

- AMI-Spezifikation für Rechenressourcen
- Tutorial: Erstellen Sie ein Compute-Ressourcen-AMI
- Verwenden Sie ein GPU-Workload-AMI
- Veraltete Amazon Linux-Version

AMI-Spezifikation für Rechenressourcen

Die grundlegende AMI-Spezifikation für AWS Batch Rechenressourcen umfasst Folgendes:

Erforderlich

- Eine moderne Linux-Distribution, auf der mindestens Version 3.10 des Linux-Kernels auf einem AMI vom Typ HVM-Virtualisierung ausgeführt wird. Windows-Container werden nicht unterstützt.

> **Important**
>
> parallel Jobs mit mehreren Knoten können nur auf Rechenressourcen ausgeführt werden, die auf einer Amazon Linux-Instance mit dem installierten `ecs-init` Paket gestartet wurden. Wir empfehlen, dass Sie bei der Erstellung Ihrer Datenverarbeitungsumgebung das standardmäßige, für Amazon ECS optimierte AMI verwenden. Sie können dies tun,

indem Sie kein benutzerdefiniertes AMI angeben. Weitere Informationen finden Sie unter parallel Jobs mit mehreren Knoten.

- Der Amazon ECS-Containeragent. Wir empfehlen, dass Sie die neueste Version verwenden. Weitere Informationen finden Sie unter Installation des Amazon ECS Container Agent im Amazon Elastic Container Service Developer Guide.
- Der `awslogs` Protokolltreiber muss als verfügbarer Protokolltreiber mit der `ECS_AVAILABLE_LOGGING_DRIVERS` Umgebungsvariablen angegeben werden, wenn der Amazon ECS-Container-Agent gestartet wird. Weitere Informationen finden Sie unter Amazon ECS Container Agent Configuration im Entwicklerhandbuch zum Amazon Elastic Container Service.
- Ein Docker-Daemon, auf dem mindestens Version 1.9 ausgeführt wird, und alle Abhängigkeiten zur Docker-Laufzeit. Weitere Informationen finden Sie unter Check runtime dependencies (Prüfen der Laufzeitabhängigkeiten) in der Docker-Dokumentation.

> **Note**
>
> Wir empfehlen die Docker-Version, die mit der entsprechenden Amazon ECS-Agentenversion, die Sie verwenden, geliefert wird und mit dieser getestet wurde. Amazon ECS stellt ein Changelog für die Linux-Variante des Amazon ECS-optimierten AMI on bereit. GitHub Weitere Informationen finden Sie unter Änderungsprotokoll.

Empfohlen

- Ein Initialisierungs- und Nanny-Prozess zur Ausführung und Überwachung des Amazon ECS-Agenten. Das für Amazon ECS optimierte AMI verwendet den `ecs-init` Upstart-Prozess, und andere Betriebssysteme verwenden `systemd` ihn möglicherweise. Weitere Informationen und Beispiele finden Sie unter Beispielskripte zur Konfiguration von Benutzerdaten für Container-Instances im Amazon Elastic Container Service Developer Guide. Weitere Informationen zu `ecs-init` finden Sie im `ecs-init`Projekt unter GitHub. Für verwaltete Rechenumgebungen muss der Amazon ECS-Agent mindestens beim Booten gestartet werden. Wenn der Amazon ECS-Agent nicht auf Ihrer Rechenressource läuft, kann er keine Jobs von annehmen AWS Batch.

Das für Amazon ECS optimierte AMI ist mit diesen Anforderungen und Empfehlungen vorkonfiguriert. Wir empfehlen, dass Sie das für Amazon ECS optimierte AMI oder ein Amazon Linux AMI mit dem `ecs-init` Paket verwenden, das für Ihre Rechenressourcen installiert ist. Wählen Sie ein anderes

AMI, wenn Ihre Anwendung ein bestimmtes Betriebssystem oder eine Docker-Version benötigt, die in diesen AMIs noch nicht verfügbar ist. Weitere Informationen finden Sie unter Amazon ECS-Optimized AMI im Amazon Elastic Container Service Developer Guide.

Tutorial: Erstellen Sie ein Compute-Ressourcen-AMI

Sie können Ihr eigenes benutzerdefiniertes Compute-Ressourcen-AMI erstellen, das Sie für Ihre verwalteten und nicht verwalteten Computerumgebungen verwenden können. Anweisungen hierzu finden Sie im AMI-Spezifikation für Rechenressourcen. Nachdem Sie ein benutzerdefiniertes AMI erstellt haben, können Sie eine Rechenumgebung erstellen, die dieses AMI verwendet, dem Sie eine Auftragswarteschlange zuordnen können. Beginnen Sie zuletzt damit, Jobs an diese Warteschlange zu senden.

So erstellen Sie ein benutzerdefiniertes Compute-Ressourcen-AMI

1. Wählen Sie ein Basis-AMI aus, von dem aus Sie beginnen möchten. Das Basis-AMI muss HVM-Virtualisierung verwenden. Das Basis-AMI kann kein Windows-AMI sein.

   > **Note**
   >
   > Das AMI, das Sie für eine Compute-Umgebung auswählen, muss der Architektur der Instance-Typen entsprechen, die Sie für die betreffende Compute-Umgebung verwenden möchten. Wenn Ihre Datenverarbeitungsumgebung beispielsweise A1-Instance-Typen verwendet, muss das von Ihnen ausgewählte Compute-Ressourcen-AMI ARM-Ressourcen unterstützen. Amazon ECS verkauft sowohl x86- als auch ARM-Versionen des für Amazon ECS optimierten Amazon Linux 2-AMI. Weitere Informationen finden Sie unter Amazon ECS-optimiertes Amazon Linux 2-AMI im Amazon Elastic Container Service Developer Guide.

   Das für Amazon ECS optimierte Amazon Linux 2-AMI ist das Standard-AMI für Rechenressourcen in verwalteten Computerumgebungen. Das für Amazon ECS optimierte Amazon Linux 2-AMI ist vorkonfiguriert und wurde AWS Batch von AWS Technikern getestet. Es handelt sich um ein minimales AMI, mit dem Sie beginnen können und mit dem Sie Ihre Rechenressourcen AWS schnell zum Laufen bringen können. Weitere Informationen finden Sie unter Amazon ECS Optimized AMI im Amazon Elastic Container Service Developer Guide.

Alternativ können Sie eine andere Amazon Linux 2-Variante wählen und das `ecs-init` Paket mit den folgenden Befehlen installieren. Weitere Informationen finden Sie unter Installation des Amazon ECS-Container-Agenten auf einer Amazon Linux EC2 2-Instance im Amazon Elastic Container Service Developer Guide:

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Wenn Sie beispielsweise GPU-Workloads auf Ihren AWS Batch Rechenressourcen ausführen möchten, können Sie mit dem Amazon Linux Deep Learning AMI beginnen. Konfigurieren Sie dann das AMI für die Ausführung von AWS Batch Jobs. Weitere Informationen finden Sie unter Verwenden Sie ein GPU-Workload-AMI.

> **Important**
>
> Sie können ein Basis-AMI wählen, das das `ecs-init` Paket nicht unterstützt. In diesem Fall müssen Sie jedoch eine Methode konfigurieren, mit der der Amazon ECS-Agent beim Booten gestartet und weiter ausgeführt werden kann. Sie können sich auch mehrere Beispielskripts zur Konfiguration von Benutzerdaten ansehen, die `systemd` zum Starten und Überwachen des Amazon ECS-Container-Agenten verwendet werden. Weitere Informationen finden Sie unter Beispiel für Konfigurationsskripte für Container-Instance-Benutzerdaten im Amazon Elastic Container Service Developer Guide.

2. Starten Sie eine Instance von Ihrem ausgewählten Basis-AMI aus mit den entsprechenden Speicheroptionen für Ihr AMI. Sie können die Größe und Anzahl der angehängten Amazon EBS-Volumes oder Instance-Speicher-Volumes konfigurieren, sofern der von Ihnen gewählte Instance-Typ diese unterstützt. Weitere Informationen finden Sie unter Launching an Instance und Amazon EC2 Instance Store im EC2 Amazon-Benutzerhandbuch.
3. Connect zu Ihrer Instance mit her SSH und führen Sie alle erforderlichen Konfigurationsaufgaben durch. Dies kann einen oder alle der folgenden Schritte beinhalten:
   - Installation des Amazon ECS-Container-Agenten. Weitere Informationen finden Sie unter Installation des Amazon ECS Container Agent im Amazon Elastic Container Service Developer Guide.
   - Konfigurieren Sie ein Skript zum Formatieren von Instance-Speicher-Volumes.

- Hinzufügen von Instance-Speicher-Volume- oder Amazon EFS-Dateisystemen zur `/etc/fstab` Datei, sodass sie beim Booten gemountet werden.
- Konfiguration von Docker-Optionen, z. B. das Aktivieren des Debuggens oder das Anpassen der Basis-Image-Größe.
- Installieren Sie Pakete oder kopieren Sie Dateien.

Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Ihrer Linux-Instance mithilfe von SSH im EC2 Amazon-Benutzerhandbuch.

4. Wenn Sie den Amazon ECS-Container-Agenten auf Ihrer Instance gestartet haben, müssen Sie ihn beenden und alle persistenten Daten-Checkpoint-Dateien entfernen, bevor Sie Ihr AMI erstellen. Andernfalls, wenn Sie dies nicht tun, startet der Agent nicht auf Instances, die von Ihrem AMI aus gestartet werden.

   a. Halten Sie den Amazon-ECS-Container-Agent an.

      - Amazon-ECS-optimiertes Amazon Linux 2-AMI:

      ```
      sudo systemctl stop ecs
      ```

      - Amazon-ECS-optimiertes Amazon Linux AMI:

      ```
      sudo stop ecs
      ```

   b. Entfernen Sie die Checkpoint-Dateien für persistente Daten. Standardmäßig befinden sich diese Dateien im `/var/lib/ecs/data/` Verzeichnis. Verwenden Sie den folgenden Befehl, um diese Dateien zu entfernen, falls vorhanden.

   ```
   sudo rm -rf /var/lib/ecs/data/*
   ```

5. Erstellen Sie ein neues AMI aus Ihrer laufenden Instance. Weitere Informationen finden Sie unter Creating an Amazon EBS Backed Linux AMI im EC2 Amazon-Benutzerhandbuch.

Um Ihr neues AMI zu verwenden mit AWS Batch

1. Nachdem das neue AMI erstellt wurde, erstellen Sie eine Rechenumgebung mit dem neuen AMI. Wählen Sie dazu den Image-Typ aus und geben Sie die benutzerdefinierte AMI-ID in das Feld Image-ID Override ein, wenn Sie die AWS Batch Rechenumgebung erstellen.

Weitere Informationen finden Sie unter the section called “Tutorial: Erstellen Sie eine verwaltete Rechenumgebung mithilfe von EC2 Amazon-Ressourcen”.

> **Note**
>
> Das AMI, das Sie für eine Compute-Umgebung auswählen, muss der Architektur der Instance-Typen entsprechen, die Sie für die betreffende Compute-Umgebung verwenden möchten. Wenn Ihre Datenverarbeitungsumgebung beispielsweise A1-Instance-Typen verwendet, muss das von Ihnen ausgewählte Compute-Ressourcen-AMI ARM-Ressourcen unterstützen. Amazon ECS verkauft sowohl x86- als auch ARM-Versionen des für Amazon ECS optimierten Amazon Linux 2-AMI. Weitere Informationen finden Sie unter Amazon ECS-optimiertes Amazon Linux 2-AMI im Amazon Elastic Container Service Developer Guide.

2. Erstellen Sie eine Auftragswarteschlange und verknüpfen Sie Ihre neue Datenverarbeitungsumgebung. Weitere Informationen finden Sie unter Eine Job-Warteschlange erstellen.

> **Note**
>
> Alle Rechenumgebungen, die mit einer Job-Warteschlange verknüpft sind, müssen dieselbe Architektur haben. AWS Batch unterstützt nicht das Mischen von Architekturtypen für Rechenumgebungen in einer einzigen Jobwarteschlange.

3. (Optional) Übermitteln Sie einen Beispielauftrag an die neue Auftragswarteschlange. Weitere Informationen finden Sie unter Referenz: Beispiele für Berufsdefinitionen, Erstellen Sie eine Auftragsdefinition mit einem Knoten und Tutorial: Einen Job einreichen.

Verwenden Sie ein GPU-Workload-AMI

Um GPU-Workloads auf Ihren AWS Batch Rechenressourcen auszuführen, müssen Sie ein AMI mit GPU-Unterstützung verwenden. Weitere Informationen finden Sie unter Arbeiten mit GPUs auf Amazon ECS und Amazon ECS-Optimized AMIs im Amazon Elastic Container Service Developer Guide.

Wenn in verwalteten Rechenumgebungen die Rechenumgebung irgendwelchep3,,,p4,p5, p6 g3 g3s g4g5, oder g6 Instance-Typen oder Instance-Familien spezifiziert, AWS Batch verwendet sie ein Amazon ECS-GPU-optimiertes AMI.

In nicht verwalteten Computerumgebungen wird ein GPU-optimiertes Amazon ECS-AMI empfohlen. Sie können die GetParametersByPathOperationen AWS Command Line Interface oder AWS Systems Manager Parameter Store GetParameter, und verwenden GetParameters, um die Metadaten für das empfohlene AMIs GPU-optimierte Amazon ECS abzurufen.

> **Note**
>
> Die p5 Instance-Familie wird nur in Versionen unterstützt, die dem Amazon ECS-GPU-optimierten AMI entsprechen oder später sind, und sie sind nicht mit p2 Instance-Typen kompatibel. 20230912 g2 Wenn Sie p5 Instances verwenden müssen, stellen Sie sicher, dass Ihre Rechenumgebung keine g2 OD-Instances enthält p2 und das neueste Standard-Batch-AMI verwendet. Beim Erstellen einer neuen Rechenumgebung wird das neueste AMI verwendet. Wenn Sie Ihre Rechenumgebung jedoch so aktualisieren, dass sie inkludiertp5, können Sie sicherstellen, dass Sie das neueste AMI verwenden, indem Sie `true` in den `ComputeResource` Eigenschaften `updateToLatestImageVersion`auf einstellen. Weitere Informationen zur AMI-Kompatibilität mit GPU-Instances finden Sie unter Working with GPUs on Amazon ECS im Amazon Elastic Container Service Developer Guide.

Die folgenden Beispiele demonstrieren die Verwendung des GetParameter-Befehls.

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended \
                        --region us-east-2 --output json
```

Die Ausgabe enthält die AMI-Informationen im `Value` Parameter.

```
{
    "Parameter": {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
        "LastModifiedDate": 1555434128.664,
        "Value": "{\"schema_version\":1,\"image_name\":\"amzn2-ami-ecs-gpu-
hvm-2.0.20190402-x86_64-ebs\",\"image_id\":\"ami-083c800fe4211192f\",\"os\":\"Amazon
 Linux 2\",\"ecs_runtime_version\":\"Docker version 18.06.1-ce\",\"ecs_agent_version
\":\"1.27.0\"}",
        "Version": 9,
        "Type": "String",
```

```
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended"
    }
}
```

Python

```
from __future__ import print_function

import json
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/
gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id   = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])
```

Die Ausgabe umfasst nur die AMI-ID und den AMI-Namen:

```
image_id   = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Die folgenden Beispiele demonstrieren die Verwendung von GetParameters.

AWS CLI

```
$ aws ssm get-parameters --names  /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_name \
                                  /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
                         --region us-east-2 --output json
```

Die Ausgabe enthält die vollständigen Metadaten für die einzelnen Parameter:

```
{
    "InvalidParameters": [],
    "Parameters": [
        {
```

```
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        }
    ]
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
            Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',
                   '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

Die Ausgabe enthält die AMI-ID und den AMI-Namen, wobei der vollständige Pfad für die Namen verwendet wird.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
 ami-083c800fe4211192f
```

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Die folgenden Beispiele demonstrieren die Verwendung des GetParametersByPath-Befehls.

AWS CLI

```
$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended \
                                --region us-east-2 --output json
```

Die Ausgabe enthält die vollständigen Metadaten für alle Parameter unter dem angegebenen Pfad.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_agent_version",
            "LastModifiedDate": 1555434128.801,
            "Value": "1.27.0",
            "Version": 8,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_agent_version"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_runtime_version",
            "LastModifiedDate": 1548368308.213,
            "Value": "Docker version 18.06.1-ce",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_runtime_version"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
            "Version": 9,
```

```
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
            "LastModifiedDate": 1548368308.143,
            "Value": "Amazon Linux 2",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
            "LastModifiedDate": 1548368307.914,
            "Value": "1",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
        }
    ]
}
```

Python

```
from __future__ import print_function

import boto3
```

```
ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

Die Ausgabe enthält die Werte aller Parameternamen im angegebenen Pfad, wobei der vollständige Pfad für die Namen verwendet wird.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =
 1.27.0
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =
 Docker version 18.06.1-ce
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
 ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Weitere Informationen finden Sie unter Abrufen von Amazon ECS-optimierten AMI-Metadaten im Amazon Elastic Container Service Developer Guide.

Veraltete Amazon Linux-Version

Das Amazon Linux AMI (auch Amazon Linux 1 genannt) hat am 31. Dezember 2023 sein Lebensende erreicht. AWS Batch hat den Support für Amazon Linux AMI eingestellt, da es ab dem 1. Januar 2024 keine Sicherheitsupdates oder Bugfixes erhalten wird. Weitere Informationen zu Amazon Linux end-of-life finden Sie in den häufig gestellten Fragen zu AL.

Wir empfehlen, bestehende Amazon Linux-basierte Rechenumgebungen auf Amazon Linux 2023 zu aktualisieren, um unvorhergesehene Workload-Unterbrechungen zu vermeiden und weiterhin Sicherheits- und andere Updates zu erhalten.

Ihre Rechenumgebungen, die das Amazon Linux AMI verwenden, funktionieren möglicherweise auch nach dem 31. Dezember 2023 end-of-life weiter. Diese Computerumgebungen erhalten jedoch keine neuen Softwareupdates, Sicherheitspatches oder Bugfixes mehr von AWS. Es liegt in Ihrer Verantwortung, diese Rechenumgebungen anschließend auf dem Amazon Linux AMI

aufrechtzuerhalten end-of-life. Wir empfehlen, AWS Batch Rechenumgebungen auf Amazon Linux 2023 oder Amazon Linux 2 zu migrieren, um optimale Leistung und Sicherheit zu gewährleisten.

Hilfe bei der Migration AWS Batch vom Amazon Linux AMI zu Amazon Linux 2023 oder Amazon Linux 2 finden Sie unter Computerumgebungen aktualisieren — AWS Batch.

Verwenden Sie EC2 Amazon-Startvorlagen mit AWS Batch

AWS Batch unterstützt die Verwendung von EC2 Amazon-Startvorlagen in Ihren EC2 Computerumgebungen. Mit Startvorlagen können Sie die Standardkonfiguration Ihrer AWS Batch Rechenressourcen ändern, ohne benutzerdefinierte AMIs Vorlagen erstellen zu müssen.

> **Note**
>
> Startvorlagen werden auf AWS Fargate-Ressourcen nicht unterstützt.

Sie müssen eine Startvorlage erstellen, bevor Sie sie mit einer Datenverarbeitungsumgebung verknüpfen können. Sie können eine Startvorlage in der EC2 Amazon-Konsole erstellen. Sie können auch das AWS CLI oder ein AWS SDK verwenden. Die folgende JSON-Datei stellt beispielsweise eine Startvorlage dar, die die Größe des Docker-Datenvolumes für das AWS Batch Standard-Compute-Ressourcen-AMI ändert und es auch so festlegt, dass es verschlüsselt wird.

```
{
    "LaunchTemplateName": "increase-container-volume-encrypt",
    "LaunchTemplateData": {
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "Encrypted": true,
                    "VolumeSize": 100,
                    "VolumeType": "gp2"
                }
            }
        ]
    }
}
```

Sie können die vorherige Startvorlage erstellen, indem Sie die JSON-Datei in einer Datei speichern, die aufgerufen wird, `lt-data.json` und den folgenden AWS CLI Befehl ausführen.

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Weitere Informationen zu Startvorlagen finden Sie unter Launching an Instance from a Launch Template im EC2 Amazon-Benutzerhandbuch.

Wenn Sie eine Startvorlage verwenden, um Ihre Datenverarbeitungsumgebung zu erstellen, können Sie die folgenden vorhandenen Parameter der Datenverarbeitungsumgebung in Ihre Startvorlage verschieben:

> **Note**
>
> Nehmen wir an, dass einer dieser Parameter (mit Ausnahme der EC2 Amazon-Tags) sowohl in der Startvorlage als auch in der Konfiguration der Rechenumgebung angegeben ist. Dann haben die Parameter der Rechenumgebung Vorrang. EC2 Amazon-Tags werden zwischen der Startvorlage und der Konfiguration der Rechenumgebung zusammengeführt. Wenn der Schlüssel des Tags kollidiert, hat der Wert in der Konfiguration der Rechenumgebung Vorrang.

- EC2 Amazon-Schlüsselpaar
- EC2 Amazon-AMI-ID
- Sicherheitsgruppe IDs
- EC2 Amazon-Tags

Die folgenden Parameter der Startvorlage werden ignoriert von AWS Batch:

- Instance-Typ (Legen Sie beim Erstellen Ihrer Datenverarbeitungsumgebung die gewünschten Instance-Typen fest.)
- Instance-Rolle (Legen Sie beim Erstellen Ihrer Datenverarbeitungsumgebung die gewünschte Instance-Rolle fest.)
- Netzwerkschnittstellen-Subnetze (Legen Sie beim Erstellen Ihrer Datenverarbeitungsumgebung die gewünschten Subnetze fest.)
- Optionen für den Instance-Markt (AWS Batch muss die Spot-Instance-Konfiguration steuern)

- Deaktivieren Sie die API-Kündigung (AWS Batch muss den Instance-Lebenszyklus kontrollieren)

AWS Batch aktualisiert die Startvorlage nur bei Infrastruktur-Updates mit einer neuen Version der Startvorlage. Weitere Informationen finden Sie unter Aktualisierung von Rechenumgebungen.

Startvorlagen als Standard festlegen und überschreiben

Sie können eine Standard-Startvorlage für die Rechenumgebung und eine Override-Startvorlage für bestimmte Instance-Typen und -Familien definieren. Dies kann für Sie nützlich sein, damit die Standardvorlage für die meisten Instance-Typen in den Rechenumgebungen verwendet wird.

Die Substitutionsvariablen `$Default` und `$Latest` können verwendet werden, anstatt eine bestimmte Version zu benennen. Wenn Sie keine überschriebene Startvorlage angeben, wird automatisch die Standard-Startvorlage angewendet.

Wenn Sie entweder die `$Latest` Variable `$Default` oder verwenden, AWS Batch werden die aktuellen Informationen zum Zeitpunkt der Erstellung der Rechenumgebung übernommen. Wenn sich die Standard- oder neueste Version in future ändert, müssen Sie die Informationen über UpdateComputeEnvironmentoder über das AWS Management Console - aktualisieren AWS Batch.

Um zusätzliche Flexibilität zu bieten, können Sie definieren, dass Override-Startvorlagen auf bestimmte Compute-Instance-Typen oder -Familien angewendet werden.

> **Note**
>
> Sie können bis zu zehn (10) Override-Startvorlagen pro Rechenumgebung angeben.

Verwenden Sie den `targetInstanceTypes` Parameter, um den Instance-Typ oder die Instance-Familie auszuwählen, die diese Override-Startvorlage verwenden soll. Der Instance-Typ oder die Instance-Familie muss zuerst durch den `instanceTypes`Parameter identifiziert werden.

Wenn Sie Überschreibungen für Startvorlagen definieren und diese später entfernen möchten, können Sie ein leeres Array übergeben, um den `overrides`Parameter im `UpdateComputeEnvironment`API-Vorgang aufzuheben. Sie können sich auch dafür entscheiden, den `overrides` Parameter beim Absenden des `UpdateComputeEnvironment` API-Vorgangs nicht einzubeziehen. Weitere Informationen finden Sie unter `LaunchTemplateSpecification.overrides`

Weitere Informationen finden Sie `LaunchTemplateSpecificationOverride.targetInstanceTypes`im AWS Batch API-Referenzhandbuch.

EC2 Amazon-Benutzerdaten in Startvorlagen

Sie können EC2 Amazon-Benutzerdaten in Ihrer Startvorlage angeben, die von cloud-init ausgeführt wird, wenn Ihre Instances gestartet werden. Mit Ihren Benutzerdaten können gängige Konfigurationsszenarien durchgeführt werden, einschließlich, aber nicht beschränkt auf die folgenden:

- Einschließen von Benutzern oder Gruppen
- Installieren von Paketen
- Erstellen von Partitionen und Dateisystemen

EC2 Amazon-Benutzerdaten in Startvorlagen müssen im mehrteiligen MIME-Archivformat vorliegen. Das liegt daran, dass Ihre Benutzerdaten mit anderen AWS Batch Benutzerdaten zusammengeführt werden, die für die Konfiguration Ihrer Rechenressourcen erforderlich sind. Sie können mehrere Benutzerdatenblöcke in einer einzelnen mehrteiligen MIME-Datei kombinieren. Beispielsweise möchten Sie vielleicht einen Cloud-Boothook, der den Docker-Daemon konfiguriert, mit einem Benutzerdaten-Shell-Skript kombinieren, das Konfigurationsinformationen für den Amazon ECS-Container-Agenten schreibt.

Wenn Sie den AWS::CloudFormation::InitTyp verwenden AWS CloudFormation, kann er zusammen mit dem Hilfsskript cfn-init verwendet werden, um allgemeine Konfigurationsszenarien durchzuführen.

Eine mehrteilige MIME-Datei umfasst folgende Komponenten:

- Deklaration von Inhaltstyp und Teilgrenze: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- Deklaration der MIME-Version: `MIME-Version: 1.0`
- Ein oder mehrere Benutzerdatenblöcke, die die folgenden Komponenten enthalten:
  - Die Öffnungsgrenze, die den Beginn eines Benutzerdatenblocks signalisiert:`--==BOUNDARY==`. Sie müssen die Zeile vor dieser Grenze leer lassen.
  - Die Inhaltstypdeklaration für den Block: `Content-Type: text/cloud-config; charset="us-ascii"`. Weitere Informationen zu Inhaltstypen finden Sie in der Cloud-Init-Dokumentation. Sie müssen die Zeile nach der Inhaltstyp-Deklaration leer lassen.

- Der Inhalt der Benutzerdaten, z. B. eine Liste von Shell-Befehlen oder `cloud-init` -Direktiven.
- Die schließende Grenze, die das Ende der mehrteiligen MIME-Datei signalisiert:`--==BOUNDARY==--`. Sie müssen die Zeile vor der schließenden Grenze leer lassen.

> **Note**
>
> Wenn Sie Benutzerdaten zu einer Startvorlage in der EC2 Amazon-Konsole hinzufügen, können Sie sie als Klartext einfügen. Oder Sie können es aus einer Datei hochladen. Wenn Sie das AWS CLI oder ein AWS SDK verwenden, müssen Sie zuerst die Benutzerdaten `base64` codieren und diese Zeichenfolge beim Aufrufen als Wert des `UserData` Parameters angeben CreateLaunchTemplate, wie in dieser JSON-Datei gezeigt.
>
> ```
> {
>  "LaunchTemplateName": "base64-user-data",
>  "LaunchTemplateData": {
>  "UserData":
>  "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
>  }
> }
> ```

Themen

- Referenz: Beispiele für EC2 Amazon-Startvorlagen

Referenz: Beispiele für EC2 Amazon-Startvorlagen

Im Folgenden finden Sie mehrteilige MIME-Beispieldateien, mit denen Sie Ihre eigenen Vorlagen erstellen können.

Beispiele

- Beispiel: Mounten Sie ein vorhandenes Amazon EFS-Dateisystem
- Beispiel: Überschreiben Sie die Standardkonfiguration des Amazon ECS-Container-Agenten
- Beispiel: Mounten Sie ein vorhandenes Amazon FSx for Lustre-Dateisystem

Beispiel: Mounten Sie ein vorhandenes Amazon EFS-Dateisystem

Example

Diese mehrteilige MIME-Beispieldatei konfiguriert die Rechenressource für die Installation des `amazon-efs-utils` Pakets und das Mounten eines vorhandenen Amazon EFS-Dateisystems unter. `/mnt/efs`

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY==="

--===MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--===MYBOUNDARY==--
```

Beispiel: Überschreiben Sie die Standardkonfiguration des Amazon ECS-Container-Agenten

Example

Diese mehrteilige MIME-Beispieldatei überschreibt die standardmäßigen Cleanup-Einstellungen des Docker-Images für eine Datenverarbeitungsressource.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY==="

--===MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
```

```
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config

--==MYBOUNDARY==--
```

Beispiel: Mounten Sie ein vorhandenes Amazon FSx for Lustre-Dateisystem

Example

In dieser mehrteiligen MIME-Beispieldatei wird die Rechenressource so konfiguriert, dass sie das `lustre2.10` Paket aus der Extras-Bibliothek installiert und ein FSx für Lustre vorhandenes Dateisystem mit dem Mount-Namen `/scratch` einhängt. `fsx` Dieses Beispiel bezieht sich auf Amazon Linux 2. Installationsanweisungen für andere Linux-Distributionen finden Sie unter Installation des Lustre-Clients im Amazon FSx for Lustre-Benutzerhandbuch. Weitere Informationen finden Sie unter Automatisches Mounten Ihres FSx Amazon-Dateisystems im Amazon FSx for Lustre-Benutzerhandbuch.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
 ${fsx_directory}

--==MYBOUNDARY==--
```

Bei den Mitgliedern volumes und mountPoints der Containereigenschaften müssen die Einhängepunkte dem Container zugeordnet sein.

```
{
    "volumes": [
        {
            "host": {
```

```
                "sourcePath": "/scratch"
            },
            "name": "Scratch"
        }
    ],
    "mountPoints": [
        {
            "containerPath": "/scratch",
            "sourceVolume": "Scratch"
        }
    ],
}
```

EC2 Konfigurationen

AWS Batch verwendet Amazon ECS, optimiert AMIs für EC2 EC2 Spot-Computing-Umgebungen. Die Standardeinstellung ist Amazon Linux 2 (`ECS_AL2`). Vor dem 31. März 2021 war dieser Standard Amazon Linux (`ECS_AL1`) für Nicht-GPU- und AWS Nicht-Graviton-Instances.

> **Note**
>
> AWS Batch unterstützt auch Amazon Linux 2023.

Das Amazon Linux AMI (auch Amazon Linux 1 genannt) hat am 31. Dezember 2023 sein Lebensende erreicht. AWS Batch hat den Support für Amazon Linux AMI eingestellt, da es ab dem 1. Januar 2024 keine Sicherheitsupdates oder Bugfixes erhalten wird. Weitere Informationen zu Amazon Linux end-of-life finden Sie in den häufig gestellten Fragen zu AL.

Wir empfehlen, bestehende Amazon Linux-basierte Rechenumgebungen auf Amazon Linux 2023 zu aktualisieren, um unvorhergesehene Workload-Unterbrechungen zu vermeiden und weiterhin Sicherheits- und andere Updates zu erhalten.

Ihre Rechenumgebungen, die das Amazon Linux AMI verwenden, funktionieren möglicherweise auch nach dem 31. Dezember 2023 end-of-life weiter. Diese Computerumgebungen erhalten jedoch keine neuen Softwareupdates, Sicherheitspatches oder Bugfixes mehr von AWS. Es liegt in Ihrer Verantwortung, diese Rechenumgebungen anschließend auf dem Amazon Linux AMI aufrechtzuerhalten end-of-life. Wir empfehlen, AWS Batch Rechenumgebungen auf Amazon Linux 2023 oder Amazon Linux 2 zu migrieren, um optimale Leistung und Sicherheit zu gewährleisten.

Hilfe bei der Migration AWS Batch vom Amazon Linux AMI zu Amazon Linux 2023 oder Amazon Linux 2 finden Sie unter Aktualisieren von Rechenumgebungen — AWS Batch

Strategien zur Zuweisung von Instance-Typen für AWS Batch

Wenn eine verwaltete Rechenumgebung erstellt wird, AWS Batch wählt sie aus den `instanceTypes` angegebenen Instanztypen aus, die den Anforderungen der Jobs am besten entsprechen. Die Zuweisungsstrategie definiert das Verhalten, wenn zusätzliche Kapazität AWS Batch benötigt wird. Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden. Geben Sie diesen Parameter nicht an.

`BEST_FIT` (Standard)

AWS Batch wählt einen Instance-Typ aus, der den Anforderungen der Jobs am besten entspricht, wobei der Instance-Typ mit den niedrigsten Kosten bevorzugt wird. Wenn zusätzliche Instanzen des ausgewählten Instanztyps nicht verfügbar sind, AWS Batch wartet es, bis die zusätzlichen Instanzen verfügbar sind. Wenn nicht genügend Instances verfügbar sind oder wenn der Benutzer die EC2 Amazon-Servicekontingente erreicht, werden zusätzliche Jobs erst ausgeführt, wenn die aktuell ausgeführten Jobs abgeschlossen sind. Diese Zuweisungsstrategie hält die Kosten niedriger, kann aber die Skalierung einschränken. Wenn Sie Spot-Flotten mit verwenden`BEST_FIT`, muss die IAM-Rolle Spot-Flotte angegeben werden. `BEST_FIT`wird bei der Aktualisierung von Rechenumgebungen nicht unterstützt. Weitere Informationen finden Sie unter Aktualisierung von Rechenumgebungen.

> **Note**
>
> AWS Batch verwaltet AWS Ressourcen in Ihrem Konto. Computerumgebungen mit der BEST_FIT-Zuweisungsstrategie verwendeten ursprünglich standardmäßig Startkonfigurationen. Die Verwendung von Startkonfigurationen mit neuen AWS Konten wird jedoch im Laufe der Zeit eingeschränkt. Daher werden neu erstellte BEST_FIT-Rechenumgebungen ab Ende April 2024 standardmäßig Vorlagen starten. Falls Ihre Servicerolle nicht über die erforderlichen Berechtigungen zur Verwaltung von Startvorlagen verfügt, AWS Batch können Sie weiterhin Startkonfigurationen verwenden. Bestehende Computerumgebungen werden weiterhin Startkonfigurationen verwenden.

`BEST_FIT_PROGRESSIVE`

AWS Batch wählt zusätzliche Instanztypen aus, die groß genug sind, um die Anforderungen der Jobs in der Warteschlange zu erfüllen. Instanztypen mit niedrigeren Kosten für jede vCPU-Einheit werden bevorzugt. Wenn zusätzliche Instanzen der zuvor ausgewählten Instanztypen nicht verfügbar sind, AWS Batch wählt neue Instanztypen aus.

> **Note**
>
> AWS Batch Wählt für parallel Jobs mit mehreren Knoten den optimalen verfügbaren Instanztyp aus. Wenn der Instance-Typ aufgrund unzureichender Kapazität nicht mehr verfügbar ist, werden andere Instance-Typen innerhalb der Familie nicht gestartet.

`SPOT_CAPACITY_OPTIMIZED`

AWS Batch wählt einen oder mehrere Instance-Typen aus, die groß genug sind, um die Anforderungen der Jobs in der Warteschlange zu erfüllen. Instanztypen, bei denen die Wahrscheinlichkeit einer Unterbrechung geringer ist, werden bevorzugt. Diese Zuweisungsstrategie ist nur für Spot-Instance-Datenverarbeitungsressourcen verfügbar.

`SPOT_PRICE_CAPACITY_OPTIMIZED`

Die preis- und kapazitätsoptimierte Zuweisungsstrategie betrachtet sowohl den Preis als auch die Kapazität, um die Spot-Instance-Pools auszuwählen, die am unwahrscheinlichsten unterbrochen werden und den niedrigstmöglichen Preis haben. Diese Zuweisungsstrategie ist nur für Spot-Instance-Datenverarbeitungsressourcen verfügbar.

> **Note**
>
> Wir empfehlen, dass Sie `SPOT_PRICE_CAPACITY_OPTIMIZED` eher als `SPOT_CAPACITY_OPTIMIZED` in den meisten Fällen verwenden.

Die `BEST_FIT` Strategien `BEST_FIT_PROGRESSIVE` und verwenden On-Demand-Instances oder Spot-Instances, `SPOT_CAPACITY_OPTIMIZED` und die `SPOT_PRICE_CAPACITY_OPTIMIZED` Strategien verwenden Spot-Instances. AWS Batch Möglicherweise müssen diese jedoch überschritten werden, `maxvCpus` um Ihre Kapazitätsanforderungen zu erfüllen. Überschreitet `maxvCpus` in diesem Fall AWS Batch nie mehr als eine Instanz.

Speicherverwaltung für Rechenressourcen

Wenn der Amazon ECS-Container-Agent eine Rechenressource in einer Rechenumgebung registriert, muss der Agent ermitteln, wie viel Speicher die Rechenressource zur Verfügung hat, um sie für Ihre Jobs zu reservieren. Aufgrund des Mehraufwands des Plattformspeichers und des vom Systemkernel belegten Speichers unterscheidet sich diese Zahl von der installierten Speichermenge für EC2 Amazon-Instances. Eine `m4.large`-Instance beispielsweise besitzt 8 GiB installierten Speicher. Dies entspricht jedoch nicht immer genau 8192 MiB Arbeitsspeicher, der für Jobs verfügbar ist, wenn die Rechenressource registriert wird.

Angenommen, Sie geben 8192 MiB für den Job an und für keine Ihrer Rechenressourcen stehen 8192 MiB oder mehr Arbeitsspeicher zur Verfügung, um diese Anforderung zu erfüllen. Dann kann der Job nicht in Ihrer Computerumgebung platziert werden. Wenn Sie eine verwaltete Rechenumgebung verwenden, AWS Batch müssen Sie einen größeren Instanztyp starten, um der Anfrage gerecht zu werden.

Das AWS Batch Standard-Rechenressourcen-AMI reserviert außerdem 32 MiB Speicher für den Amazon ECS-Container-Agenten und andere kritische Systemprozesse. Dieser Speicher ist für die Auftragszuweisung nicht verfügbar. Weitere Informationen finden Sie unter Systemspeicher reservieren.

Der Amazon-ECS-Container-Agent verwendet die Docker-Funktion `ReadMemInfo()` für die Abfrage des gesamten für das Betriebssystem verfügbaren Arbeitsspeichers. Linux stellt Befehlszeilenprogramme zur Verfügung, um den Gesamtspeicher zu ermitteln.

Example – Bestimmung des Gesamtspeichers für Linux

Der free Befehl gibt den Gesamtspeicher zurück, der vom Betriebssystem erkannt wird.

```
$ free -b
```

Im Folgenden finden Sie eine Beispielausgabe für eine `m4.large` Instance, auf der das Amazon ECS-optimierte Amazon Linux AMI ausgeführt wird.

```
             total       used       free     shared    buffers     cached
Mem:    8373026816  348180480 8024846336      90112   25534464  205418496
-/+ buffers/cache:  117227520 8255799296
```

Diese Instance verfügt über einen Gesamtspeicher von 8373026816 Byte. Das bedeutet, dass 7985 MiB für Aufgaben zur Verfügung stehen.

Themen

- Systemspeicher reservieren
- Tutorial: Rechenressourcenspeicher anzeigen
- Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS

Systemspeicher reservieren

Wenn Sie mit Ihren Jobs den gesamten Arbeitsspeicher einer Rechenressource belegen, ist es möglich, dass Ihre Jobs mit kritischen Systemprozessen um Arbeitsspeicher zu kämpfen haben und möglicherweise zu einem Systemausfall führen. Der Amazon ECS-Container-Agent stellt eine Konfigurationsvariable bereit, die aufgerufen wird`ECS_RESERVED_MEMORY`. Sie können diese Konfigurationsvariable verwenden, um eine bestimmte Anzahl von MiB Speicher aus dem Pool zu entfernen, der Ihren Jobs zugewiesen ist. Damit wird dieser Arbeitsspeicher für wichtige Systemprozesse reserviert.

Das AWS Batch Standard-Rechenressourcen-AMI reserviert 32 MiB Arbeitsspeicher für den Amazon ECS-Container-Agenten und andere kritische Systemprozesse.

Tutorial: Rechenressourcenspeicher anzeigen

Sie können in der Amazon ECS-Konsole oder bei der DescribeContainerInstancesAPI-Operation sehen, mit wie viel Speicher sich eine Rechenressource registriert. Wenn Sie versuchen, Ihre Ressourcennutzung zu maximieren, indem Sie Ihren Jobs so viel Speicher wie möglich für einen bestimmten Instance-Typ zur Verfügung stellen, können Sie den für diese Rechenressource verfügbaren Speicher beobachten und Ihren Jobs dann so viel Speicher zuweisen.

Um den Arbeitsspeicher der Rechenressource anzuzeigen

1. Öffnen Sie die Konsole auf https://console.aws.amazon.com/ecs/Version 2.
2. Wählen Sie Cluster und dann den Cluster aus, der Ihre anzuzeigenden Rechenressourcen hostet.

   Der Cluster-Name für Ihre Computing-Umgebung beginnt mit dem Namen Ihrer Computing-Umgebung.
3. Wählen Sie Infrastruktur aus.
4. Wählen Sie unter Container-Instances die Container-Instance aus.

5. Im Abschnitt Ressourcen und Netzwerke wird der registrierte und verfügbare Speicher für die Rechenressource angezeigt.

   Der Wert für registrierten Speicher ist der Wert, den die Rechenressource bei Amazon ECS registriert hat, als sie zum ersten Mal gestartet wurde, und der Wert für verfügbaren Speicher ist der Wert, der noch keinen Aufträgen zugewiesen wurde.

Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS

In AWS Batch Amazon EKS können Sie die Ressourcen angeben, die einem Container zur Verfügung gestellt werden. Sie können beispielsweise `limits` OR-Werte für vCPU- und Speicherressourcen angeben`requests`.

Die folgenden Einschränkungen gelten für die Angabe von vCPU-Ressourcen:

- Es muss mindestens eine vCPU `requests` oder ein `limits` Wert angegeben werden.
- Eine vCPU-Einheit entspricht einem physischen oder virtuellen Kern.
- Der vCPU-Wert muss in ganzen Zahlen oder in Schritten von 0,25 eingegeben werden.
- Der kleinste gültige vCPU-Wert ist 0,25.
- Wenn beide angegeben sind, muss der `requests` Wert kleiner oder gleich dem `limits` Wert sein. Auf diese Weise können Sie sowohl weiche als auch harte vCPU-Konfigurationen konfigurieren.
- vCPU-Werte können nicht in MilliCPU-Form angegeben werden. `100m`Ist beispielsweise kein gültiger Wert.
- AWS Batch verwendet den `requests` Wert für Skalierungsentscheidungen. Wenn kein `requests` Wert angegeben ist, wird der `limits` Wert in den `requests` Wert kopiert.

Die folgenden Einschränkungen gelten für die Angabe von Speicherressourcen:

- Es muss mindestens ein Speicher `requests` oder `limits` Wert angegeben werden.
- Speicherwerte müssen sich in befinden mebibytes (MiBs).
- Wenn beide angegeben sind, muss der `requests` Wert dem `limits` Wert entsprechen.
- AWS Batch verwendet den `requests` Wert für Skalierungsentscheidungen. Wenn kein `requests` Wert angegeben ist, wird der `limits` Wert in den `requests` Wert kopiert.

Die folgenden Einschränkungen gelten für die Angabe von GPU-Ressourcen:

- Wenn beide angegeben sind, muss der `requests` Wert dem `limits` Wert entsprechen.
- AWS Batch verwendet den `requests` Wert für Skalierungsentscheidungen. Wenn kein `requests` Wert angegeben ist, wird der `limits` Wert in den `requests` Wert kopiert.

Beispiel: Jobdefinitionen

Mit der folgenden Jobdefinition in AWS Batch Amazon EKS werden Soft-vCPU-Shares konfiguriert. Dadurch kann AWS Batch Amazon EKS die gesamte vCPU-Kapazität für den Instance-Typ nutzen. Wenn jedoch andere Jobs ausgeführt werden, wird dem Job ein Maximum von 2 v CPUs zugewiesen. Der Arbeitsspeicher ist auf 2 GB begrenzt.

```
{
    "jobDefinitionName": "MyJobOnEks_Sleep",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "requests": {
                            "cpu": "2",
                            "memory": "2048Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

Die folgende Auftragsdefinition in AWS Batch Amazon EKS hat einen `request` Wert von 1 und weist dem Auftrag ein Maximum von 4 v CPUs zu.

```
{
    "jobDefinitionName": "MyJobOnEks_Sleep",
    "type": "container",
    "eksProperties": {
```

```
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "requests": {
                            "cpu": "1"
                        },
                        "limits": {
                            "cpu": "4",
                            "memory": "2048Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

Die folgende Auftragsdefinition in AWS Batch Amazon EKS legt einen `limits` vCPU-Wert von 1 und einen `limits` Speicherwert von 1 GB fest.

```
{
    "jobDefinitionName": "MyJobOnEks_Sleep",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

Wenn ein Job AWS Batch auf Amazon EKS in einen Amazon EKS-Pod AWS Batch übersetzt wird, wird der `limits` Wert in den `requests` Wert AWS Batch kopiert. Dies ist der Fall, wenn kein `requests` Wert angegeben ist. Wenn Sie die obige Beispiel-Jobdefinition einreichen, sieht der Pod wie `spec` folgt aus.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
        - sleep
        - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
        requests:
          cpu: 1
          memory: 1024Mi
      ...
```

CPU- und Speicherreservierungen für Knoten

AWS Batch stützt sich auf die Standardlogik der `bootstrap.sh` Datei für vCPU- und Speicherreservierungen. Weitere Informationen zur `bootstrap.sh` Datei finden Sie unter bootstrap.sh. Beachten Sie bei der Dimensionierung Ihrer vCPU- und Speicherressourcen die folgenden Beispiele.

> **Note**
>
> Wenn keine Instances ausgeführt werden, können sich vCPU- und Speicherreservierungen zunächst auf die AWS Batch Skalierungslogik und die Entscheidungsfindung auswirken. AWS Batch Passt die anfänglichen Zuweisungen an, nachdem die Instanzen ausgeführt wurden.

Beispiel: CPU-Reservierung eines Knotens

Der CPU-Reservierungswert wird in Millikernen berechnet, wobei die Gesamtzahl der V verwendet wirdCPUs , die für die Instanz verfügbar sind.

vCPU-Nummer	Prozentsatz reserviert
1	6%
2	1%
3-4	0.5%
4 und höher	0,25%

Unter Verwendung der vorherigen Werte ist Folgendes wahr:

- Der CPU-Reservierungswert für eine `c5.large` Instance mit 2 v CPUs ist 70 m. Dies wird auf folgende Weise berechnet: (1*60) + (1*10) = 70 m.
- Der CPU-Reservierungswert für eine `c5.24xlarge` Instanz mit 96 v beträgt 310 m. CPUs Dies wird auf folgende Weise berechnet: (1*60) + (1*10) + (2*5) + (92*2,5) = 310 m.

In diesem Beispiel stehen 1930 (berechnete 2000-70) Millicore-vCPU-Einheiten zur Ausführung von Jobs auf einer Instance zur Verfügung. `c5.large` Angenommen, Ihr Job erfordert 2 (2*1000 m) vCPU-Einheiten, der Job passt nicht auf eine einzelne Instanz. `c5.large` Ein Job, der `1.75` vCPU-Einheiten erfordert, passt jedoch.

Beispiel: Speicherreservierung für Knoten

Der Wert für die Speicherreservierung wird wie folgt in Mebibyte berechnet:

- Die Instanzkapazität in Mebibyte. Eine 8-GB-Instance ist beispielsweise 7.748 MiB.
- Der `kubeReserved` Wert. Der `kubeReserved` Wert ist die Speichermenge, die für System-Daemons reserviert werden soll. Der `kubeReserved` Wert wird auf folgende Weise berechnet: ((11 * maximale Anzahl von Pods, die vom Instance-Typ unterstützt wird) + 255). Informationen zur maximalen Anzahl von Pods, die von einem Instance-Typ unterstützt werden, finden Sie unter eni-max-pods.txt

- Der `HardEvictionLimit` Wert. Wenn der verfügbare Speicher unter den `HardEvictionLimit` Wert fällt, versucht die Instance, Pods zu entfernen.

Die Formel zur Berechnung des zuweisbaren Speichers lautet wie folgt:
(*`instance_capacity_in_MiB`*) - (11 * (*`maximum_number_of_pods`*)) - 255 - ()).
`HardEvictionLimit value.`

Eine `c5.large` Instance unterstützt bis zu 29 Pods. Für eine `c5.large` 8-GB-Instance mit einem `HardEvictionLimit` Wert von 100 MiB beträgt der zuweisbare Speicher 7074 MiB. Dies wird wie folgt berechnet: (7748 - (11 * 29) -255 -100) = 7074 MiB. In diesem Beispiel sind es 8.192 MiB Job passt nicht auf diese Instanz, obwohl es eine 8 ist gibibyte (GiB) Instanz.

DaemonSets

Wenn du benutzt DaemonSets, beachten Sie Folgendes:

- Wenn AWS Batch auf Amazon EKS keine Instances laufen, DaemonSets kann sich zunächst auf die AWS Batch Skalierungslogik und die Entscheidungsfindung auswirken. AWS Batch weist zunächst 0,5 vCPU-Einheiten und 500 MiB für die erwarteten Werte zu DaemonSets. AWS Batch Passt die anfänglichen Zuweisungen an, nachdem die Instances ausgeführt wurden.
- Wenn ein DaemonSet definiert vCPU- oder Speicherlimits, bei AWS Batch Amazon EKS haben Jobs weniger Ressourcen. Wir empfehlen Ihnen, die Anzahl der beizubehalten DaemonSets die AWS Batch Jobs so gering wie möglich zugewiesen sind.

Aktualisierung von Rechenumgebungen

Nachdem Sie eine Rechenumgebung erstellt haben, die EC2 Ressourcen verwendet, können Sie viele Einstellungen der Rechenumgebung direkt aktualisieren. Das Ändern einiger Einstellungen erfordert jedoch, dass die Instanzen in der Rechenumgebung AWS Batch ersetzt werden.

> **Important**
>
> AWS Batch erstellt und verwaltet mehrere AWS Ressourcen in Ihrem Namen und innerhalb Ihres Kontos, darunter Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets und Amazon ECS-Cluster. Diese verwalteten Ressourcen sind speziell konfiguriert, um einen optimalen AWS Batch Betrieb zu gewährleisten. Die manuelle Änderung dieser im Batch verwalteten Ressourcen kann, sofern nicht ausdrücklich in der

AWS Batch Dokumentation angegeben, zu unerwartetem Verhalten führen, das zu einer `INVALID` Rechenumgebung, suboptimalem Instanzskalierungsverhalten, verzögerter Workload-Verarbeitung oder unerwarteten Kosten führt. Diese manuellen Änderungen können vom Service nicht deterministisch unterstützt werden. AWS Batch Verwenden Sie immer die unterstützte Batch APIs - oder Batch-Konsole, um Ihre Rechenumgebungen zu verwalten.

AWS Fargate Computerumgebungen aktualisieren

Für Rechenumgebungen, die Fargate-Ressourcen verwenden, können Sie Folgendes aktualisieren.

- `securityGroupIds`
- `subnets`
- `desiredvCpus`
- `maxvCpus`
- `minvCpus`

AWS Batch hat zwei Aktualisierungsmechanismen. Das erste ist ein Skalierungsupdate, bei dem Instanzen zur Rechenumgebung hinzugefügt oder aus ihr entfernt werden. Das zweite ist ein Infrastruktur-Update, bei dem die Instanzen in der Rechenumgebung ersetzt werden. Ein Infrastruktur-Update dauert viel länger als ein Skalierungsupdate.

Wenn Sie Rechenumgebungen mit aktualisieren AWS Batch, führt eine Änderung nur dieser Einstellungen zu einem Skalierungsupdate: gewünschtes v CPUs (`desiredvCpus`), maximales v CPUs (`maxvCpus`), minimales v CPUs (`minvCpus`), dienstrolle (`serviceRole`) und state (`state`).

> **Note**
>
> Wenn Sie die `desiredvCpus` Einstellung aktualisieren, muss der Wert zwischen den `maxvCpus` Werten `minvCpus` und liegen.
> Darüber hinaus muss der aktualisierte `desiredvCpus` Wert größer oder gleich dem aktuellen `desiredvCpus` Wert sein. Weitere Informationen finden Sie unter the section called “Fehlermeldung beim Aktualisieren der `desiredvCpus` Einstellung”.

Wenn eine der folgenden Einstellungen in einem geändert wird UpdateComputeEnvironmentAPI-Aktion, AWS Batch initiiert ein Infrastruktur-Update. Ein Infrastruktur-Update erfordert, dass die Servicerolle auf AWSServiceRoleForBatch(Standard) gesetzt ist und dass die ZuweisungsstrategieBEST_FIT_PROGRESSIVE,SPOT_CAPACITY_OPTIMIZED, oder SPOT_PRICE_CAPACITY_OPTIMIZED lautet. BEST_FITwird nicht unterstützt. Mit Ausnahme der Servicerolle können alle Einstellungen, die für ein Skalierungsupdate geändert werden können, auch für ein Infrastruktur-Update geändert werden.

> **Note**
>
> SPOT_CAPACITY_OPTIMIZEDIn den meisten Fällen empfehlen wir die Verwendung SPOT_PRICE_CAPACITY_OPTIMIZED von.

Während eines Infrastruktur-Updates ändert sich der Status der Rechenumgebung aufUPDATING. Neue Instanzen werden mit den aktualisierten Einstellungen gestartet. Neue Jobs sind für die neuen Instanzen geplant. Jobs, die derzeit ausgeführt werden, werden gemäß der Richtlinie für Infrastrukturaktualisierungen versandt. Weitere Informationen finden Sie unter UpdateComputeEnvironment und UpdatePolicy in der AWS Batch -API-Referenz.

Berücksichtigen Sie beim UpdatePolicy Datentyp die folgenden Szenarien:

> **Note**
>
> In diesen Szenarien gilt Folgendes. Wenn eine Instanz beendet wird, werden laufende Jobs gestoppt. Standardmäßig werden diese Jobs nicht erneut versucht. Um einen dieser Jobs nach dem Beenden einer Instance erneut zu versuchen, konfigurieren Sie eine Strategie zur Wiederholung von Aufträgen. Weitere Informationen finden Sie unter the section called “Automatisierte Auftragswiederholungen” im AWS Batch -Benutzerhandbuch.

- Wenn die terminateJobsOnUpdate Einstellung auf gesetzt isttrue, werden laufende Jobs während eines Infrastruktur-Updates beendet. Die jobExecutionTimeoutMinutes Einstellung wird ignoriert.
- Wenn die terminateJobsOnUpdate Einstellung auf gesetzt istfalse, können Jobs nach dem Infrastruktur-Update für zusätzliche Zeit ausgeführt werden. Diese zusätzliche Zeit ist in der jobExecutionTimeoutMinutes Einstellung konfiguriert. Die jobExecutionTimeoutMinutes Standardeinstellung ist 30 Minuten.

Sobald Kapazität in der Rechenumgebung verfügbar wird, werden neue Instanzen mit den aktualisierten Einstellungen gestartet und Jobs auf den neuen Instanzen gestartet. Sobald alle Jobs auf Instanzen mit den alten Einstellungen abgeschlossen sind, werden die alten Instanzen beendet. Wenn Kapazität verfügbar wird, bedeutet dies, dass die gewünschte Anzahl von v um mindestens so viele v unter der maximalen Anzahl CPUs von v CPUs liegt, CPUs wie es für den kleinsten Instance-Typ erforderlich ist.

Aktualisierungen der Infrastruktur

Ein Infrastruktur-Update ist erforderlich, um einige Einstellungen für eine Rechenumgebung zu ändern. Wenn eine der folgenden Einstellungen geändert wird, wird ein Infrastruktur-Update gestartet:

> **Important**
>
> Die Rechenumgebung muss die AWSServiceRoleForBatchserviceverknüpfte Rolle verwenden, um Änderungen vorzunehmen, die ein Infrastruktur-Update erfordern.
> Wenn die Rechenumgebung eine serviceverknüpfte Rolle verwendet, kann sie nicht geändert werden, sodass sie eine reguläre IAM-Rolle verwendet. Wenn die Rechenumgebung über eine reguläre IAM-Rolle verfügt, kann diese ebenfalls nicht so geändert werden, dass sie eine dienstverknüpfte Rolle verwendet. Daher können Sie Infrastrukturaktualisierungen nur für Rechenumgebungen durchführen, die mithilfe einer serviceverknüpften Rolle erstellt wurden.

- Die Zuweisungsstrategie (`allocationStrategy`, muss entweder `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, oder `SPOT_PRICE_CAPACITY_OPTIMIZED` lauten. Wenn die ursprüngliche Zuweisungsstrategie lautet`BEST_FIT`, werden Infrastrukturaktualisierungen nicht unterstützt.)

  > **Note**
  >
  > `SPOT_CAPACITY_OPTIMIZED`In den meisten Fällen empfehlen wir`SPOT_PRICE_CAPACITY_OPTIMIZED`, diese Option zu verwenden.

- Prozentsatz des Gebots (`bidPercentage`)
- EC2 Konfiguration (`ec2Configuration`)
- Schlüsselpaar (`ec2KeyPair`)
- Bild-ID (`imageId`)

- Instanzrolle (`instanceRole`)
- Instanztypen (`instanceTypes`)
- Vorlage starten (`launchTemplate`)
- Platzierungsgruppe (`placementGroup`)
- Sicherheitsgruppen (`securityGroupIds`)
- VPC-Subnetze () `subnets`
- EC2 Schlagworte () `tags`
- Typ der Rechenumgebung (`type`, kann einer von `EC2` oder sein`SPOT`)
- Ob AWS Batch während eines Infrastruktur-Updates auf das neueste AMI aktualisiert werden soll, das von unterstützt wird `updateToLatestImageVersion`

Aktualisierung der AMI-ID

Während eines Infrastruktur-Updates kann sich die AMI-ID der Rechenumgebung ändern, je nachdem, ob AMIs sie in einer dieser drei Einstellungen angegeben ist. AMIs sind in der `imageId` (in`computeResources`), `imageIdOverride` (in`ec2Configuration`) oder der unter angegebenen Startvorlage angegeben`launchTemplate`. Nehmen wir an, dass in keiner dieser Einstellungen AMI angegeben IDs sind und die `updateToLatestImageVersion` Einstellung ist`true`. Anschließend wird das neueste Amazon ECS-optimierte AMI, das von unterstützt AWS Batch wird, für jedes Infrastruktur-Update verwendet.

Wenn in mindestens einer dieser Einstellungen eine AMI-ID angegeben ist, hängt das Update davon ab, welche Einstellung die vor dem Update verwendete AMI-ID bereitgestellt hat. Wenn Sie eine Rechenumgebung erstellen, ist die Priorität für die Auswahl einer AMI-ID zuerst die Startvorlage, dann die `imageId` Einstellung und schließlich die `imageIdOverride` Einstellung. Wenn die verwendete AMI-ID jedoch aus der Startvorlage stammt, wird durch das Aktualisieren der `imageId` `imageIdOverride` Oder-Einstellungen die AMI-ID nicht aktualisiert. Die einzige Möglichkeit, eine aus der Startvorlage ausgewählte AMI-ID zu aktualisieren, besteht darin, die Startvorlage zu aktualisieren. Wenn der Versionsparameter der Startvorlage `$Default` oder ist`$Latest`, wird die Standard- oder neueste Version der angegebenen Startvorlage ausgewertet. Wenn standardmäßig eine andere AMI-ID ausgewählt ist oder die neueste Version der Startvorlage ausgewählt ist, wird diese AMI-ID im Update verwendet.

Wenn die Startvorlage nicht zur Auswahl der AMI-ID verwendet wurde, wird die AMI-ID verwendet, die in den `imageIdOverride` Parametern `imageId` oder angegeben ist. Wenn beide angegeben sind, wird die im `imageIdOverride` Parameter angegebene AMI-ID verwendet.

Angenommen, die Rechenumgebung verwendet eine AMI-ID`imageId`, die durch die `launchTemplate` Parameter`imageIdOverride`, oder angegeben wird, und Sie möchten das neueste für Amazon ECS optimierte AMI verwenden, das von unterstützt wird AWS Batch. Anschließend muss das Update die Einstellungen entfernen, die AMI bereitgestellt haben IDs. Denn `imageId` dazu muss eine leere Zeichenfolge für diesen Parameter angegeben werden. Denn `imageIdOverride` dazu muss eine leere Zeichenfolge für den `ec2Configuration` Parameter angegeben werden.

Wenn die AMI-ID aus der Startvorlage stammt, können Sie zum neuesten Amazon ECS-optimierten AMI wechseln, das auf eine der folgenden Arten unterstützt wird: AWS Batch

- Entfernen Sie die Startvorlage, indem Sie eine leere Zeichenfolge für den `launchTemplateName` Parameter `launchTemplateId` oder angeben. Dadurch wird die gesamte Startvorlage entfernt und nicht nur die AMI-ID.
- Wenn die aktualisierte Version der Startvorlage keine AMI-ID angibt, muss der `updateToLatestImageVersion` Parameter auf gesetzt werden`true`.

Fargate-Computerumgebungen

Fargate ist eine Technologie, mit der Sie Container ausführen können AWS Batch , ohne Server oder Cluster von EC2 Amazon-Instances verwalten zu müssen. Mit Fargate müssen Sie keine Cluster virtueller Maschinen mehr bereitstellen, konfigurieren oder skalieren, um Container auszuführen. Auf diese Weise müssen keine Servertypen mehr ausgewählt werden, es muss nicht entschieden werden, wann die Cluster skaliert werden oder das Cluster-Packing optimiert werden.

Wenn Sie Ihre Jobs mit Fargate-Ressourcen ausführen, verpacken Sie Ihre Anwendung in Containern, geben die CPU- und Speicheranforderungen an, definieren Netzwerk- und IAM-Richtlinien und starten die Anwendung. Jeder Fargate-Job hat seine eigene Isolationsgrenze und teilt den zugrunde liegenden Kernel, die CPU-Ressourcen, Speicherressourcen oder die elastic network interface nicht mit einem anderen Job.

Themen

- Wann sollte Fargate verwendet werden

- Jobdefinitionen bei Fargate
- Warteschlangen für Job auf Fargate
- Rechenumgebungen auf Fargate

Wann sollte Fargate verwendet werden

Wir empfehlen die Verwendung von Fargate in den meisten Szenarien. Fargate startet und skaliert die Berechnung so, dass sie den Ressourcenanforderungen, die Sie für den Container angeben, genau entspricht. Mit Fargate müssen Sie nicht zu viele Server bereitstellen oder für zusätzliche Server bezahlen. Sie müssen sich auch keine Gedanken über die Besonderheiten infrastrukturbezogener Parameter wie den Instance-Typ machen. Wenn die Rechenumgebung skaliert werden muss, können Jobs, die auf Fargate-Ressourcen ausgeführt werden, schneller gestartet werden. In der Regel dauert es einige Minuten, eine neue EC2 Amazon-Instance hochzufahren. Jobs, die auf Fargate ausgeführt werden, können jedoch in etwa 30 Sekunden bereitgestellt werden. Die genaue benötigte Zeit hängt von mehreren Faktoren ab, darunter der Größe des Container-Images und der Anzahl der Jobs.

Wir empfehlen Ihnen jedoch, Amazon zu verwenden, EC2 wenn Ihre Jobs eine der folgenden Voraussetzungen erfordern:

- Mehr als 16 v CPUs
- Mehr als 120 Gibibyte (GiB) Speicher
- EINE GPU
- Ein benutzerdefiniertes Amazon Machine Image (AMI)
- Jeder der LinuxParameter-Parameter

Wenn Sie eine große Anzahl von Jobs haben, empfehlen wir Ihnen, die EC2 Amazon-Infrastruktur zu verwenden. Zum Beispiel, wenn die Anzahl der gleichzeitig ausgeführten Jobs die Drosselungsgrenzen von Fargate überschreitet. Dies liegt daran, dass Aufträge mit EC2 einer höheren Geschwindigkeit an EC2 Ressourcen als an Fargate-Ressourcen weitergeleitet werden können. Außerdem können bei Verwendung mehr Jobs gleichzeitig ausgeführt werden. EC2 Weitere Informationen finden Sie unter Fargate Service Quotas im Amazon Elastic Container Service Developer Guide.

Jobdefinitionen bei Fargate

AWS Batch jobs on unterstützt AWS Fargate nicht alle verfügbaren Jobdefinitionsparameter. Einige Parameter werden überhaupt nicht unterstützt, und andere verhalten sich für Fargate-Jobs anders.

In der folgenden Liste werden Jobdefinitionsparameter beschrieben, die in Fargate-Aufträgen nicht gültig oder anderweitig eingeschränkt sind.

`platformCapabilities`

Muss als `FARGATE` angegeben werden.

```
"platformCapabilities": [ "FARGATE" ]
```

`type`

Muss als angegeben werden`container`.

```
"type": "container"
```

Parameter in `containerProperties`

`executionRoleArn`

Muss für Jobs angegeben werden, die auf Fargate-Ressourcen ausgeführt werden. Weitere Informationen finden Sie unter IAM-Rollen für Aufgaben im Entwicklerhandbuch zum Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

`fargatePlatformConfiguration`

(Optional, nur für Fargate-Jobdefinitionen). Gibt die Fargate-Plattformversion oder `LATEST` für eine aktuelle Plattformversion an. Mögliche Werte für `platformVersion` sind `1.3.0`1`.4.0`, und `LATEST` (Standard).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

`instanceType`, `ulimits`

Gilt nicht für Jobs, die auf Fargate-Ressourcen ausgeführt werden.

`memory`, `vcpus`

Diese Einstellungen müssen in angegeben werden `resourceRequirements`

`privileged`

Geben Sie diesen Parameter entweder nicht an oder geben Sie an`false`.

```
"privileged": false
```

`resourceRequirements`

Sowohl die Speicher- als auch die vCPU-Anforderungen müssen mit unterstützten Werten angegeben werden. GPU-Ressourcen werden für Jobs, die auf Fargate-Ressourcen ausgeführt werden, nicht unterstützt.

Wenn Sie GuardDuty Runtime Monitoring verwenden, entsteht ein geringer Speicheraufwand für den GuardDuty Security Agent. Daher muss das Speicherlimit die Größe des GuardDuty Security Agents beinhalten. Informationen zu den Speicherlimits des GuardDuty Security Agents finden Sie unter CPU- und Speicherlimits im GuardDuty Benutzerhandbuch. Informationen zu den bewährten Methoden finden Sie im Amazon ECS-Entwicklerhandbuch unter Wie behebe ich Fehler bei Speichermangel bei meinen Fargate-Aufgaben nach der Aktivierung von Runtime Monitoring?.

```
"resourceRequirements": [
  {"type": "MEMORY", "value": "512"},
  {"type": "VCPU",   "value": "0.25"}
]
```

Parameter in `linuxParameters`

`devices`, `maxSwap`, `sharedMemorySize`, `swappiness`, `tmpfs`

Gilt nicht für Jobs, die auf Fargate-Ressourcen ausgeführt werden.

Parameter in `logConfiguration`

`logDriver`

Es werden ausschließlich `awslogs` und `splunk` unterstützt. Weitere Informationen finden Sie unter Verwenden Sie den awslogs-Protokolltreiber.

Mitglieder in `networkConfiguration`

`assignPublicIp`

Wenn an das private Subnetz kein NAT-Gateway angeschlossen ist, um Datenverkehr ins Internet zu senden, `assignPublicIp` muss "`ENABLED`" angegeben werden. Weitere Informationen finden Sie unter AWS Batch IAM-Ausführungsrolle.

Warteschlangen für Job auf Fargate

AWS Batch Die Jobwarteschlangen an AWS Fargate sind im Wesentlichen unverändert. Die einzige Einschränkung besteht darin, dass die Rechenumgebungen, die in aufgeführt sind, alle Fargate-Rechenumgebungen (`FARGATE`oder`FARGATE_SPOT`) sein `computeEnvironmentOrder` müssen. EC2 und Fargate-Rechenumgebungen können nicht gemischt werden.

Rechenumgebungen auf Fargate

AWS Batch Rechenumgebungen, die aktiviert sind, unterstützen AWS Fargate nicht alle verfügbaren Parameter der Rechenumgebung. Einige Parameter werden überhaupt nicht unterstützt. Andere haben spezielle Anforderungen für Fargate.

In der folgenden Liste werden Computing-Umgebungsparameter beschrieben, die in Fargate-Jobs nicht gültig oder anderweitig eingeschränkt sind.

`type`

Dieser Parameter muss sein. `MANAGED`

```
"type": "MANAGED"
```

Parameter im `computeResources` Objekt

`allocationStrategy`, `bidPercentage`, `desiredvCpus`, `imageId`, `instanceTypes`, `ec2Configuration`, `ec2KeyPair`, `instanceRole`, `launchTemplate`, `minvCpus`, `placementGroup`, `spotIamFleetRole`

Diese gelten nicht für Fargate-Rechenumgebungen und können nicht bereitgestellt werden.

`subnets`

Wenn an die in diesem Parameter aufgeführten Subnetze keine NAT-Gateways angeschlossen sind, muss der `assignPublicIp` Parameter in der Jobdefinition auf gesetzt werden. `ENABLED`

`tags`

Dies gilt nicht für Fargate-Rechenumgebungen und kann nicht bereitgestellt werden. Um Tags für Fargate-Rechenumgebungen anzugeben, verwenden Sie den `tags` Parameter, der nicht im `computeResources` Objekt enthalten ist.

`type`

Muss entweder `FARGATE` oder `FARGATE_SPOT` sein.

```
"type": "FARGATE_SPOT"
```

Amazon EKS-Rechenumgebungen

Erste Schritte mit AWS Batch Amazon EKSbietet eine kurze Anleitung zur Erstellung von EKS-Rechenumgebungen. Dieser Abschnitt enthält weitere Informationen zu Amazon EKS-Rechenumgebungen.

AWS Batch vereinfacht Ihre Batch-Workloads auf Amazon EKS-Clustern durch die Bereitstellung verwalteter Batch-Funktionen. Dazu gehören Warteschlangen, Abhängigkeitsverfolgung, verwaltete Auftragswiederholungen und Prioritäten, Pod-Verwaltung und Knotenskalierung. AWS Batch kann mehrere Availability Zones und mehrere EC2 Amazon-Instance-Typen und -Größen verarbeiten. AWS Batch integriert mehrere der Best Practices von Amazon EC2 Spot, um Ihre Workloads fehlertolerant auszuführen und so weniger Unterbrechungen zu ermöglichen. Sie können AWS Batch damit problemlos eine Handvoll Jobs über Nacht oder Millionen von geschäftskritischen Jobs ausführen.

AWS Batch ist ein verwalteter Service, der Batch-Workloads in Ihren Kubernetes Clustern orchestriert, die von Amazon Elastic Kubernetes Service (Amazon EKS) verwaltet werden. AWS Batch führt diese Orchestrierung außerhalb Ihrer Cluster mithilfe eines „Overlay“ -Modells durch. Da es AWS Batch sich um einen verwalteten Dienst handelt, müssen keine Kubernetes Komponenten (z. B. Operatoren oder benutzerdefinierte Ressourcen) in Ihrem Cluster installiert oder verwaltet werden. AWS Batch Ihr Cluster muss lediglich mit rollenbasierten Zugriffskontrollen (RBAC) konfiguriert sein, die die Kommunikation mit dem AWS Batch API-Server ermöglichen. Kubernetes AWS Batch Aufrufe Kubernetes APIs zum Erstellen, Überwachen und Löschen Kubernetes von Pods und Knoten.

AWS Batch verfügt über eine integrierte Skalierungslogik zur Skalierung von Kubernetes Knoten auf der Grundlage der Auslastung der Auftragswarteschlangen mit Optimierungen im Hinblick auf die Zuweisung von Job-Kapazitäten. Wenn die Auftragswarteschlange leer ist, werden die Knoten

auf die von Ihnen festgelegte Mindestkapazität AWS Batch herunterskaliert, die standardmäßig Null ist. AWS Batch verwaltet den gesamten Lebenszyklus dieser Knoten und schmückt die Knoten mit Beschriftungen und Markierungen. Auf diese Weise werden andere Kubernetes Workloads nicht auf die Knoten übertragen, die von verwaltet werden. AWS Batch Ausgenommen hiervon sind Knoten`DaemonSets`, die auf AWS Batch Knoten abzielen können, um Überwachungs- und andere Funktionen bereitzustellen, die für die ordnungsgemäße Ausführung der Jobs erforderlich sind. Außerdem werden AWS Batch keine Jobs, insbesondere Pods, auf Knoten in Ihrem Cluster ausgeführt, die nicht verwaltet werden. Auf diese Weise können Sie separate Skalierungslogik und Dienste für andere Anwendungen im Cluster verwenden.

Um Jobs an zu senden AWS Batch, interagieren Sie direkt mit der AWS Batch API. AWS Batch übersetzt Jobs in `podspecs` und erstellt dann die Anfragen zum Platzieren von Pods auf Knoten, die von AWS Batch in Ihrem Amazon EKS-Cluster verwaltet werden. Sie können Tools verwenden, `kubectl` um beispielsweise laufende Pods und Knoten anzuzeigen. Wenn ein Pod seine Ausführung abgeschlossen hat, AWS Batch löscht er den Pod, den er erstellt hat, um die Kubernetes Systemlast zu verringern.

Sie können beginnen, indem Sie einen gültigen Amazon EKS-Cluster mit verbinden AWS Batch. Hängen Sie dann eine AWS Batch Auftragswarteschlange an und registrieren Sie eine Amazon EKS-Auftragsdefinition mit `podspec` entsprechenden Attributen. Senden Sie zuletzt Jobs mithilfe des SubmitJobAPI-Vorgangs, der auf die Auftragsdefinition verweist. Weitere Informationen finden Sie unter Erste Schritte mit AWS Batch Amazon EKS.

Amazon EKS

Themen

- AWS Batch Standard-AMI
- Gemischte AMI-Umgebungen
- Unterstützte Kubernetes-Versionen
- Aktualisieren Sie die Kubernetes Version der Rechenumgebung
- Gemeinsame Verantwortung der Kubernetes Knoten
- Führen Sie eine DaemonSet auf AWS Batch verwalteten Knoten aus
- Amazon EKS-Startvorlagen anpassen
- Wie führe ich ein Upgrade von EKS AL2 auf EKS 023 AL2 durch

AWS Batch Standard-AMI

Wenn Sie eine Amazon EKS-Rechenumgebung erstellen, müssen Sie kein Amazon Machine Image (AMI) angeben. AWS Batch wählt ein für Amazon EKS optimiertes AMI aus, das auf der Kubernetes Version und den Instance-Typen basiert, die in Ihrer CreateComputeEnvironmentAnfrage angegeben sind. Im Allgemeinen empfehlen wir, die Standard-AMI-Auswahl zu verwenden. Weitere Informationen zu Amazon EKS Optimized AMIs finden Sie unter Amazon EKS optimized Amazon Linux AMIs im Amazon EKS-Benutzerhandbuch.

Führen Sie den folgenden Befehl aus, um zu sehen, welcher AMI-Typ AWS Batch für Ihre Amazon EKS-Rechenumgebung ausgewählt wurde. Das folgende Beispiel ist ein Instance-Typ ohne GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.32 AMI,
 depending on instance types
    $ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
        | jq '.computeEnvironments[].computeResources.ec2Configuration'
    [
      {
        "imageType": "EKS_AL2",
        "imageKubernetesVersion": "1.32"
      }
    ]
```

Das folgende Beispiel ist ein GPU-Instanztyp.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.32 AMI
    $ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
        | jq '.computeEnvironments[].computeResources.ec2Configuration'
    [
      {
        "imageType": "EKS_AL2_NVIDIA",
        "imageKubernetesVersion": "1.32"
      }
    ]
```

Gemischte AMI-Umgebungen

Sie können Startvorlagen-Overrides verwenden, um Rechenumgebungen sowohl mit Amazon Linux 2 (AL2) als auch mit Amazon Linux 2023 (AL2023) zu erstellen. AMIs Dies ist nützlich, wenn Sie verschiedene Architekturen AMIs für unterschiedliche Architekturen verwenden möchten

Beispielkonfiguration mit beiden AMI-Typen:

```
{
  "computeResources": {
    "launchTemplate": {
      "launchTemplateId": "TemplateId",
      "version": "1",
      "userdataType": "EKS_BOOTSTRAP_SH",
      "overrides": [
        {
          "instanceType": "c5.large",
          "imageId": "ami-al2-custom",
          "userdataType": "EKS_BOOTSTRAP_SH"
        },
        {
          "instanceType": "c6a.large",
          "imageId": "ami-al2023-custom",
          "userdataType": "EKS_NODEADM"
        }
      ]
    },
    "instanceTypes": ["c5.large", "c6a.large"]
  }
}
```

Unterstützte Kubernetes-Versionen

AWS Batch auf Amazon unterstützt EKS derzeit die folgenden Kubernetes Versionen:

- `1.32`
- `1.31`
- `1.30`
- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`

Möglicherweise wird Ihnen eine Fehlermeldung angezeigt, die der folgenden ähnelt, wenn Sie den `CreateComputeEnvironment` API-Vorgang oder `UpdateComputeEnvironment` den API-Vorgang verwenden, um eine Rechenumgebung zu erstellen oder zu aktualisieren. Dieses Problem tritt auf, wenn Sie in `EC2Configuration` eine nicht unterstützte Kubernetes Version angeben.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Um dieses Problem zu beheben, löschen Sie die Rechenumgebung und erstellen Sie sie dann mit einer unterstützten Kubernetes Version neu.

Sie können ein kleineres Versions-Upgrade auf Ihrem Amazon EKS-Cluster durchführen. Sie können den Cluster beispielsweise von `1.xx` auf aktualisieren, `1.yy` auch wenn die Nebenversion nicht unterstützt wird.

Der Status der Rechenumgebung kann sich jedoch `INVALID` nach einem Update der Hauptversion auf ändern. Dies ist beispielsweise der Fall, wenn Sie ein Upgrade einer Hauptversion von `1.xx` auf durchführen`2.yy`. Wenn die Hauptversion von nicht unterstützt wird AWS Batch, wird eine Fehlermeldung angezeigt, die der folgenden ähnelt.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Aktualisieren Sie die Kubernetes Version der Rechenumgebung

Mit AWS Batch können Sie die Kubernetes Version einer Rechenumgebung aktualisieren, um Amazon EKS-Cluster-Upgrades zu unterstützen. Die Kubernetes Version einer Rechenumgebung ist die Amazon EKS AMI-Version für die Kubernetes Knoten, die AWS Batch gestartet wird, um Jobs auszuführen. Sie können ein Kubernetes Versions-Upgrade auf ihren Amazon EKS-Knoten durchführen, bevor oder nachdem Sie die Version der Steuerungsebene des Amazon EKS-Clusters aktualisiert haben. Wir empfehlen, die Knoten nach dem Upgrade der Kontrollebene zu aktualisieren. Weitere Informationen finden Sie unter Aktualisieren einer Amazon Kubernetes EKS-Cluster-Version im Amazon EKS-Benutzerhandbuch.

Verwenden Sie den UpdateComputeEnvironmentAPI-Vorgang, um die Kubernetes Version einer Rechenumgebung zu aktualisieren.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
```

```
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Gemeinsame Verantwortung der Kubernetes Knoten

Die Wartung der Rechenumgebungen ist eine gemeinsame Verantwortung.

- Ändern oder entfernen Sie keine AWS Batch Knoten, Labels, Taints, Namespaces, Startvorlagen oder Auto Scaling-Gruppen. Fügen Sie verwalteten Knoten keine Makel hinzu. AWS Batch Wenn Sie eine dieser Änderungen vornehmen, kann Ihre Rechenumgebung nicht unterstützt werden und es kommt zu Ausfällen, einschließlich Instanzen im Leerlauf.
- Richten Sie Ihre Pods nicht auf AWS Batch verwaltete Knoten aus. Wenn Sie Ihre Pods auf die verwalteten Knoten ausrichten, kommt es zu einer unterbrochenen Skalierung und zu festgefahrenen Jobwarteschlangen. Führen Sie Workloads aus, die nicht AWS Batch auf selbstverwalteten Knoten oder verwalteten Knotengruppen verwendet werden. Weitere Informationen finden Sie unter Verwaltete Knotengruppen im Amazon-EKS-Benutzerhandbuch.
- Sie können festlegen, DaemonSet dass a auf AWS Batch verwalteten Knoten ausgeführt wird. Weitere Informationen finden Sie unter Führen Sie eine DaemonSet auf AWS Batch verwalteten Knoten aus.

AWS Batch aktualisiert die Rechenumgebung nicht automatisch AMIs. Es liegt in Ihrer Verantwortung, sie zu aktualisieren. Führen Sie den folgenden Befehl aus, AMIs um Ihre AMI-Version auf die neueste Version zu aktualisieren.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch aktualisiert die Kubernetes Version nicht automatisch. Führen Sie den folgenden Befehl aus, um die Kubernetes Version Ihrer Computerumgebung auf zu aktualisieren*1.32*.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Bei der Aktualisierung auf ein neueres AMI oder die neuere Kubernetes Version können Sie angeben, ob Jobs beendet werden sollen, wenn sie aktualisiert werden (`terminateJobsOnUpdate`), und wie

lange gewartet werden soll, bis eine Instance ersetzt wird, wenn laufende Jobs nicht abgeschlossen werden (`jobExecutionTimeoutMinutes`.) Weitere Informationen finden Sie unter Aktualisierung von Rechenumgebungen und in der Richtlinie zur Aktualisierung der Infrastruktur (UpdatePolicy), die im UpdateComputeEnvironmentAPI-Vorgang festgelegt wurde.

Führen Sie eine DaemonSet auf AWS Batch verwalteten Knoten aus

AWS Batch setzt Fehler auf AWS Batch verwalteten Kubernetes Knoten fest. Mit den folgenden `tolerations` Optionen können Sie DaemonSet festlegen, dass a auf AWS Batch verwalteten Knoten ausgeführt wird.

```
tolerations:
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
```

Eine andere Möglichkeit, dies zu tun, ist die folgende`tolerations`.

```
tolerations:
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
    effect: "NoSchedule"
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
    effect: "NoExecute"
```

Amazon EKS-Startvorlagen anpassen

AWS Batch auf Amazon unterstützt EKS Startvorlagen. Es gibt Einschränkungen in Bezug darauf, was Ihre Startvorlage leisten kann.

> **Important**
>
> - Für EKS AL2 AMIs AWS Batch läuft`/etc/eks/bootstrap.sh`. Führen Sie es nicht `/etc/eks/bootstrap.sh` in Ihrer Startvorlage oder Ihren cloud-init user-data Skripten aus. Sie können neben dem Parameter weitere `--kubelet-extra-args` Parameter zu bootstrap.sh hinzufügen. Stellen Sie dazu die `AWS_BATCH_KUBELET_EXTRA_ARGS` Variable in der `/etc/aws-batch/batch.config` Datei ein. Einzelheiten finden Sie im folgenden Beispiel.

- Für EKS AL2 023 wird NodeConfigSpecvon EKS AWS Batch verwendet, um Instanzen zum EKS-Cluster hinzuzufügen. AWS Batch wird NodeConfigSpecfür ClusterDetailsden EKS-Cluster aufgefüllt und Sie müssen sie nicht angeben.

> **Note**
>
> Wir empfehlen, dass Sie keine der folgenden NodeConfigSpecEinstellungen in der Startvorlage festlegen, da AWS Batch diese Ihre Werte überschreiben würden. Weitere Informationen finden Sie unter Gemeinsame Verantwortung der Kubernetes Knoten.
>
> - `Taints`
> - `Cluster Name`
> - `apiServerEndpoint`
> - `certificatAuthority`
> - `CIDR`
> - Erstellen Sie keine Labels mit dem Präfix `batch.amazonaws.com/`

> **Note**
>
> Wenn die Startvorlage nach dem Aufruf geändert CreateComputeEnvironmentwird, UpdateComputeEnvironmentmuss sie aufgerufen werden, um zu überprüfen, welche Version der Startvorlage ersetzt werden kann.

Themen

- Fügen Sie kubelet zusätzliche Argumente hinzu
- Konfigurieren Sie die Container-Laufzeit
- Bereitstellen eines Amazon EFS-Volumes
- IPv6 Unterstützung

Fügen Sie **kubelet** zusätzliche Argumente hinzu

AWS Batch unterstützt das Hinzufügen zusätzlicher Argumente zum `kubelet` Befehl. Eine Liste der unterstützten Parameter finden Sie kubeletin der KubernetesDokumentation. Im folgenden Beispiel für EKS AL2 AMIs *`--node-labels mylabel=helloworld`* wird der `kubelet` Befehlszeile hinzugefügt.

```
MIME-Version: 1.0
      Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

      --==MYBOUNDARY==
      Content-Type: text/x-shellscript; charset="us-ascii"

      #!/bin/bash
      mkdir -p /etc/aws-batch

      echo AWS_BATCH_KUBELET_EXTRA_ARGS=\"--node-labels mylabel=helloworld\" >> /etc/
aws-batch/batch.config

      --==MYBOUNDARY==--
```

Für EKS AL2 023 ist AMIs das Dateiformat YAML. Eine Liste der unterstützten Parameter finden Sie NodeConfigSpecin der Kubernetes Dokumentation. Im folgenden Beispiel für EKS AL2 023 AMIs *`--node-labels mylabel=helloworld`* wird der `kubelet` Befehlszeile hinzugefügt.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: application/node.eks.aws

apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  kubelet:
    flags:
    - --node-labels=mylabel=helloworld

--==MYBOUNDARY==--
```

Konfigurieren Sie die Container-Laufzeit

Sie können die AWS Batch `CONTAINER_RUNTIME` Umgebungsvariable verwenden, um die Container-Laufzeit auf einem verwalteten Knoten zu konfigurieren. Im folgenden Beispiel wird die Container-Laufzeit auf „`containerd`when `bootstrap.sh` runs“ festgelegt. Weitere Informationen finden Sie `containerd`in der KubernetesDokumentation.

Wenn Sie ein optimiertes `EKS_AL2023` oder `EKS_AL2023_NVIDIA` AMI verwenden, müssen Sie die Container-Laufzeit nicht angeben, da nur containerd unterstützt wird.

> **Note**
>
> Die `CONTAINER_RUNTIME` Umgebungsvariable entspricht der `--container-runtime` Option von`bootstrap.sh`. Weitere Informationen finden Sie Optionsin der KubernetesDokumentation.

```
MIME-Version: 1.0
      Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

      --==MYBOUNDARY==
      Content-Type: text/x-shellscript; charset="us-ascii"

      #!/bin/bash
      mkdir -p /etc/aws-batch

      echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

      --==MYBOUNDARY==--
```

Bereitstellen eines Amazon EFS-Volumes

Sie können Startvorlagen verwenden, um Volumes auf dem Knoten zu mounten. Im folgenden Beispiel werden die `runcmd` Einstellungen `cloud-config` `packages` und verwendet. Weitere Informationen finden Sie in der cloud-initDokumentation unter Cloud-Konfigurationsbeispielen.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
```

```
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
 >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

--==MYBOUNDARY==--
```

Um dieses Volume im Job zu verwenden, muss es im eksProperties-Parameter hinzugefügt werden. RegisterJobDefinition Das folgende Beispiel ist ein großer Teil der Auftragsdefinition.

```
{
    "jobDefinitionName": "MyJobOnEks_EFS",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["ls", "-la", "/efs"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi"
                        }
                    },
                    "volumeMounts": [
                        {
                            "name": "efs-volume",
                            "mountPath": "/efs"
                        }
                    ]
                }
            ],
            "volumes": [
```

```
                {
                    "name": "efs-volume",
                    "hostPath": {
                        "path": "/mnt/efs"
                    }
                }
            ]
        }
    }
}
```

Im Knoten ist das Amazon EFS-Volume im `/mnt/efs` Verzeichnis gemountet. Im Container für den Amazon EKS-Job wird das Volume im `/efs` Verzeichnis bereitgestellt.

IPv6 Unterstützung

AWS Batch unterstützt Amazon EKS-Cluster IPv6 mit Adressen. Für den AWS Batch Support sind keine Anpassungen erforderlich. Bevor Sie beginnen, empfehlen wir Ihnen jedoch, die Überlegungen und Bedingungen zu lesen, die unter Zuweisen von IPv6 Adressen zu Pods und Services im Amazon EKS-Benutzerhandbuch beschrieben sind.

Wie führe ich ein Upgrade von EKS AL2 auf EKS 023 AL2 durch

Die für Amazon EKS optimierten Produkte AMIs sind in zwei Familien erhältlich, die auf Amazon Linux 2 (AL2) und Amazon Linux 2023 (AL2023) basieren. AL2023 ist ein neues Linux-basiertes Betriebssystem, das eine sichere, stabile und leistungsstarke Umgebung für Ihre Cloud-Anwendungen bietet. Weitere Informationen zu den Unterschieden zwischen AL2 und AL2 0.23 finden Sie unter Upgrade von Amazon Linux 2 auf Amazon Linux 2023 im Amazon EKS-Benutzerhandbuch.

Je nachdem, wie Ihre Computerumgebung konfiguriert ist, können Sie einen der folgenden Upgrade-Pfade von AL2 bis AL2 023 verwenden.

Führen Sie ein Upgrade mithilfe von Ec2Configuration durch. ImageType

- Wenn Sie keine Startvorlage oder Startvorlagen-Überschreibungen verwenden, ändern Sie Ec2Configuration. ImageTypezu `EKS_AL2023` oder und der Lauf`EKS_AL2023_NVIDIA`. UpdateComputeEnvironment
- Wenn Sie eine Ec2Configuration angeben. ImageIdOverridedann Ec2Configuration. ImageTypemuss dem in Ec2Configuration angegebenen AMI-Typ entsprechen. ImageIdOverride.

Wenn Sie nicht übereinstimmen `ImageIdOverride` und `ImageType` der Knoten dem Cluster nicht beitritt.

Führen Sie ein Upgrade mithilfe von Startvorlagen durch

- Wenn Sie `kubelet` zusätzliche Argumente in einer Startvorlage oder einer Startvorlagen-Überschreibung definiert haben, müssen diese auf das neue Format `kubelet` für zusätzliche Argumente aktualisiert werden.

  Wenn das Format der `kubelet` zusätzlichen Argumente nicht übereinstimmt, werden die zusätzlichen Argumente nicht angewendet.
- Für AL2 023 AMIs ist containerd die einzige unterstützte Container-Laufzeit. Sie müssen die Container-Laufzeit für `EKS_AL2023` in der Startvorlage nicht angeben.

  Sie können keine benutzerdefinierte Container-Laufzeit mit angeben`EKS_AL2023`.
- Wenn Sie eine Startvorlage oder eine Startvorlagenüberschreibung verwenden, die ein darauf basierendes AMI spezifiziert, `EKS_AL2023` müssen Sie userDataType auf setzen. `EKS_NODEADM`

  Wenn Sie das `userdataType` und AMI nicht übereinstimmen, tritt der Knoten dem EKS-Cluster nicht bei.

Warteschlangen für Job

Jobs werden an eine Auftragswarteschlange weitergeleitet, in der sie gespeichert werden, bis sie für die Ausführung in einer Rechenumgebung geplant werden können. Ein AWS Konto kann mehrere Jobwarteschlangen haben. Sie können beispielsweise eine Warteschlange erstellen, die Amazon EC2 On-Demand-Instances für Jobs mit hoher Priorität verwendet, und eine weitere Warteschlange, die Amazon EC2 Spot-Instances für Jobs mit niedriger Priorität verwendet. Job-Warteschlangen haben eine Priorität, die vom Scheduler verwendet wird, um zu bestimmen, welche Jobs in welcher Warteschlange zuerst zur Ausführung ausgewertet werden sollen.

Themen

- Eine Job-Warteschlange erstellen
- Status der Auftragswarteschlange anzeigen
- Fair-Share-Terminplanungsrichtlinien
- Ressourcenbewusste Planung

Eine Job-Warteschlange erstellen

Bevor Sie Jobs einreichen können AWS Batch, müssen Sie eine Auftragswarteschlange erstellen. Wenn Sie eine Auftragswarteschlange erstellen, ordnen Sie der Warteschlange eine oder mehrere Rechenumgebungen zu und weisen ihnen eine Prioritätsreihenfolge zu.

Sie legen auch der Auftragswarteschlange Priorität fest, die die Reihenfolge bestimmt, in der der AWS Batch-Scheduler Jobs platziert. Das bedeutet, dass, wenn eine Rechenumgebung mit mehr als einer Job-Warteschlange verknüpft ist, die Job-Warteschlange mit höherer Priorität bevorzugt wird.

Themen

- Tutorial: Erstellen Sie eine EC2 Amazon-Job-Warteschlange
- Tutorial: Eine Fargate-Jobwarteschlange erstellen
- Tutorial: Erstellen Sie eine Amazon EKS-Jobwarteschlange
- Referenz: Vorlage für die Job-Warteschlange

Tutorial: Erstellen Sie eine EC2 Amazon-Job-Warteschlange

Gehen Sie wie folgt vor, um eine Job-Warteschlange für Amazon Elastic Compute Cloud (Amazon EC2) zu erstellen.

Um eine EC2 Amazon-Job-Warteschlange zu erstellen

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich die Option Job-Warteschlangen aus.
4. Wählen Sie Create (Erstellen) aus.
5. Wählen Sie als Orchestrierungstyp Amazon Elastic Compute Cloud (Amazon EC2).
6. Geben Sie unter Name einen eindeutigen Namen für Ihre Job-Warteschlange ein. Der Name kann bis zu 128 Zeichen lang sein und Groß- und Kleinbuchstaben, Zahlen und Unterstriche (_) enthalten.
7. Geben Sie unter Priorität eine ganze Zahl für die Priorität der Auftragswarteschlange ein. Jobwarteschlangen mit höherer Priorität werden vor Jobwarteschlangen mit niedrigerer Priorität ausgeführt, die derselben Rechenumgebung zugeordnet sind. Die Priorität wird in absteigender Reihenfolge ermittelt. So erhält beispielsweise eine Auftragswarteschlange mit einem Prioritätswert von 10 Vorrang vor einer Auftragswarteschlange mit dem Wert 1.
8. (Optional) Wählen Sie für die Planungsrichtlinie Amazon Resource Name (ARN) eine bestehende Planungsrichtlinie aus.
9. Wählen Sie für Verbundene Rechenumgebungen eine oder mehrere Rechenumgebungen aus der Liste aus, die Sie der Job-Warteschlange zuordnen möchten. Wählen Sie die Rechenumgebungen in der Reihenfolge aus, in der die Warteschlange versuchen soll, Aufgaben zu platzieren. Der Job Scheduler bestimmt anhand der Reihenfolge, in der Sie die Rechenumgebungen auswählen, welche Rechenumgebung einen bestimmten Job startet. Bevor Sie sie einer Job-Warteschlange zuordnen können, müssen sich die Rechenumgebungen im entsprechenden `VALID` Status befinden. Sie können bis zu drei Datenverarbeitungsumgebungen mit einer Auftragswarteschlange verknüpfen. Wenn Sie noch keine Rechenumgebung haben, wählen Sie Rechenumgebung erstellen

> **Note**
>
> Alle Rechenumgebungen, die mit einer Jobwarteschlange verknüpft sind, müssen dasselbe Bereitstellungsmodell verwenden. AWS Batch unterstützt nicht das Mischen von Bereitstellungsmodellen in einer einzigen Jobwarteschlange.

10. Wählen Sie unter Reihenfolge der Rechenumgebung die Aufwärts- und Abwärtspfeile aus, um die gewünschte Reihenfolge zu konfigurieren.
11. Wählen Sie „Auftragswarteschlange erstellen“, um den Vorgang abzuschließen und Ihre Auftragswarteschlange zu erstellen.

Tutorial: Eine Fargate-Jobwarteschlange erstellen

Gehen Sie wie folgt vor, um eine Job-Warteschlange für AWS Fargate zu erstellen.

So erstellen Sie eine Fargate-Jobwarteschlange

1. Öffnen Sie die AWS Batch Konsole unter. https://console.aws.amazon.com/batch/
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich die Option Job-Warteschlangen aus.
4. Wählen Sie Create (Erstellen) aus.
5. Wählen Sie als Orchestrierungstyp Fargate aus.
6. Geben Sie unter Name einen eindeutigen Namen für Ihre Job-Warteschlange ein. Der Name kann bis zu 128 Zeichen lang sein und Groß- und Kleinbuchstaben, Zahlen und Unterstriche (_) enthalten.
7. Geben Sie unter Priorität eine ganze Zahl für die Priorität der Auftragswarteschlange ein. Jobwarteschlangen mit höherer Priorität werden vor Jobwarteschlangen mit niedrigerer Priorität ausgeführt, die derselben Rechenumgebung zugeordnet sind. Die Priorität wird in absteigender Reihenfolge ermittelt. So erhält beispielsweise eine Auftragswarteschlange mit einem Prioritätswert von 10 Vorrang vor einer Auftragswarteschlange mit dem Wert 1.
8. (Optional) Wählen Sie für die Planungsrichtlinie Amazon Resource Name (ARN) eine bestehende Planungsrichtlinie aus.
9. Wählen Sie für Verbundene Rechenumgebungen eine oder mehrere Rechenumgebungen aus der Liste aus, die Sie der Job-Warteschlange zuordnen möchten. Wählen Sie die

Rechenumgebungen in der Reihenfolge aus, in der die Warteschlange versuchen soll, Aufgaben zu platzieren. Der Job Scheduler bestimmt anhand der Reihenfolge, in der Sie die Rechenumgebungen auswählen, welche Rechenumgebung einen bestimmten Job startet. Bevor Sie sie einer Job-Warteschlange zuordnen können, müssen sich die Rechenumgebungen im entsprechenden `VALID` Status befinden. Sie können bis zu drei Datenverarbeitungsumgebungen mit einer Auftragswarteschlange verknüpfen.

> **Note**
>
> Alle Rechenumgebungen, die einer Auftragswarteschlange zugeordnet sind, müssen dasselbe Bereitstellungsmodell verwenden. AWS Batch unterstützt nicht das Mischen von Bereitstellungsmodellen in einer einzigen Jobwarteschlange.

10. Wählen Sie unter Reihenfolge der Rechenumgebung die Aufwärts- und Abwärtspfeile aus, um die gewünschte Reihenfolge zu konfigurieren.
11. Wählen Sie „Auftragswarteschlange erstellen“, um den Vorgang abzuschließen und Ihre Auftragswarteschlange zu erstellen.

Tutorial: Erstellen Sie eine Amazon EKS-Jobwarteschlange

Gehen Sie wie folgt vor, um eine Job-Warteschlange für Amazon Elastic Kubernetes Service (Amazon EKS) zu erstellen.

So erstellen Sie eine Amazon EKS-Jobwarteschlange

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich die Option Job-Warteschlangen aus.
4. Wählen Sie Create (Erstellen) aus.
5. Wählen Sie als Orchestrierungstyp Amazon Elastic Kubernetes Service (Amazon EKS) aus.
6. Geben Sie unter Name einen eindeutigen Namen für Ihre Job-Warteschlange ein. Der Name kann bis zu 128 Zeichen lang sein und Groß- und Kleinbuchstaben, Zahlen und Unterstriche (_) enthalten.
7. Geben Sie für Priorität einen Ganzzahlwert für die Priorität der Auftragswarteschlange ein. Jobwarteschlangen mit höherer Priorität werden vor Jobwarteschlangen mit niedrigerer Priorität ausgeführt, die derselben Rechenumgebung zugeordnet sind. Die Priorität wird in

absteigender Reihenfolge ermittelt. So erhält beispielsweise eine Auftragswarteschlange mit einem Prioritätswert von 10 Vorrang vor einer Auftragswarteschlange mit dem Wert 1.

8. (Optional) Wählen Sie für die Planungsrichtlinie Amazon Resource Name (ARN) eine bestehende Planungsrichtlinie aus.
9. Wählen Sie für Verbundene Rechenumgebungen eine oder mehrere Rechenumgebungen aus der Liste aus, die Sie der Job-Warteschlange zuordnen möchten. Wählen Sie die Rechenumgebungen in der Reihenfolge aus, in der die Warteschlange versuchen soll, Aufgaben zu platzieren. Der Job Scheduler bestimmt anhand der Reihenfolge, in der Sie die Rechenumgebungen auswählen, welche Rechenumgebung einen bestimmten Job startet. Bevor Sie sie einer Job-Warteschlange zuordnen können, müssen sich die Rechenumgebungen im entsprechenden `VALID` Status befinden. Sie können bis zu drei Datenverarbeitungsumgebungen mit einer Auftragswarteschlange verknüpfen.

> **Note**
>
> Alle Rechenumgebungen, die einer Auftragswarteschlange zugeordnet sind, müssen dasselbe Bereitstellungsmodell verwenden. AWS Batch unterstützt nicht das Mischen von Bereitstellungsmodellen in einer einzigen Jobwarteschlange.

> **Note**
>
> Alle Rechenumgebungen, die einer Job-Warteschlange zugeordnet sind, müssen dieselbe Architektur haben. AWS Batch unterstützt nicht das Mischen von Architekturtypen für Rechenumgebungen in einer einzigen Jobwarteschlange.

10. Wählen Sie unter Reihenfolge der Rechenumgebung die Aufwärts- und Abwärtspfeile aus, um die gewünschte Reihenfolge zu konfigurieren.
11. Wählen Sie „Auftragswarteschlange erstellen“, um den Vorgang abzuschließen und Ihre Auftragswarteschlange zu erstellen.

Referenz: Vorlage für die Job-Warteschlange

Das Folgende ist eine leere Vorlage für eine Job-Warteschlange. Sie können diese Vorlage verwenden, um Ihre Job-Warteschlange zu erstellen. Sie können diese Job-Warteschlange dann in einer Datei speichern und sie zusammen mit der AWS CLI `--cli-input-json` Option verwenden.

Weitere Informationen zu diesen Parametern finden Sie CreateJobQueuein der AWS Batch API-Referenz.

> **Note**
>
> Mit dem folgenden AWS CLI Befehl können Sie eine Vorlage für eine Job-Warteschlange generieren.
>
> ```
> $ aws batch create-job-queue --generate-cli-skeleton
> ```

```
{
   "computeEnvironmentOrder": [
      {
         "computeEnvironment": "",
         "order": 0
      }
   ],
   "jobQueueName": "",
   "jobStateTimeLimitActions": [
      {
         "state": "RUNNABLE",
         "action": "CANCEL",
   "maxTimeSeconds": 0,
         "reason": ""

      }
   ],
   "priority": 0,
   "schedulingPolicyArn": "",
   "state": "ENABLED",
   "tags": {
      "KeyName": ""
   }
}
```

Status der Auftragswarteschlange anzeigen

Nachdem Sie eine Auftragswarteschlange erstellt und die Jobs eingereicht haben, ist es wichtig, deren Fortschritt überwachen zu können. Sie können die Seite mit den Auftragsdetails verwenden, um Ihre Job-Warteschlange zu überprüfen, zu verwalten und zu überwachen.

Informationen zur Job-Warteschlange anzeigen

Wählen Sie in der AWS Batch Konsole im Navigationsbereich Auftragswarteschlangen und wählen Sie die gewünschte Auftragswarteschleife aus, um deren Details anzuzeigen. Auf dieser Seite können Sie Ihre Auftragswarteschlange überprüfen und verwalten und zusätzliche Informationen zu den Vorgängen der Warteschlange einsehen, z. B. den Job-Warteschlangen-Snapshot, die Auftragsstatusgrenzen, die Reihenfolge der Umgebung, Tags und den JSON-Code der Job-Warteschlange.

Details zur Auftragswarteschlange

Dieser Abschnitt bietet eine Übersicht und Wartungsoptionen für die Auftragswarteschlange. Es ist wichtig zu beachten, dass Sie den Amazon-Ressourcennamen (ARN) in diesem Abschnitt finden.

Um diese Informationen über zu finden AWS Command Line Interface, verwenden Sie den `DescribeJobQueues`Vorgang zusammen mit dem Namen der Auftragswarteschlange oder dem entsprechenden ARN.

Snapshot der Job-Warteschlange

Dieser Abschnitt enthält eine statische Liste der ersten 100 `RUNNABLE` Jobs, die sich in der Warteschlange befinden. Sie können das Suchfeld verwenden, um die Liste einzugrenzen, indem Sie in einer beliebigen Spalte des Ergebnisabschnitts nach Informationen suchen. Die Jobs im Bereich mit den Snapshot-Ergebnissen sind entsprechend der Ausführungsstrategie der Job-Warteschlange sortiert. Bei first-in-first-out (FIFO-) Jobwarteschlangen basiert die Reihenfolge der Jobs auf der Übermittlungszeit. Bei Jobwarteschlangen mit Fair-Share-Scheduling basiert die Reihenfolge der Jobs auf der Priorität der Jobs und der gemeinsamen Nutzung.

Da es sich bei den Ergebnissen um eine Momentaufnahme der Auftragswarteschlange handelt, wird die Ergebnisliste nicht automatisch aktualisiert. Um die Liste zu aktualisieren, wählen Sie oben im Abschnitt die Option Aktualisieren aus. Wählen Sie den Hyperlink mit dem Jobnamen, um zu den Jobdetails zu navigieren und den Status des Jobs und andere zugehörige Informationen einzusehen.

Um diese Informationen über zu finden AWS CLI, verwenden Sie den `GetJobQueueSnapshot`Vorgang zusammen mit dem Namen der Auftragswarteschlange oder dem entsprechenden ARN.

Grenzwerte für den Beschäftigungsstatus

Verwenden Sie diese Registerkarte, um die Konfigurationsinformationen darüber zu überprüfen, wie lange ein Job in einem bestimmten `RUNNABLE` Status verbleiben kann, bevor er storniert wird.

Um diese Informationen über zu finden AWS CLI, verwenden Sie den `DescribeJobQueues`Vorgang zusammen mit dem Namen der Auftragswarteschlange oder dem entsprechenden ARN.

Reihenfolge der Umgebung

Wenn Ihre Job-Warteschlange in mehreren Umgebungen läuft, finden Sie auf dieser Registerkarte deren Reihenfolge und einen Überblick.

Um diese Informationen über zu finden AWS CLI, verwenden Sie den `DescribeJobQueues`Vorgang zusammen mit dem Namen der Auftragswarteschlange oder dem entsprechenden ARN.

Tags

Verwenden Sie diese Registerkarte, um die Tags zu überprüfen und zu verwalten, die dieser Auftragswarteschlange zugeordnet sind.

JSON

Verwenden Sie diese Registerkarte, um den JSON-Code zu kopieren, der dieser Job-Warteschlange zugeordnet ist. Sie können den JSON dann für AWS CloudFormation Vorlagen und AWS CLI Skripte wiederverwenden.

Fair-Share-Terminplanungsrichtlinien

Der AWS Batch Scheduler bewertet, wann, wo und wie Jobs ausgeführt werden, die in eine Job-Warteschlange gestellt werden. Wenn Sie beim Erstellen einer Job-Warteschlange keine Scheduling-Richtlinie angeben, verwendet der AWS Batch Job-Scheduler standardmäßig eine FIFO-Strategie (First-In, First Out). Eine FIFO-Strategie kann dazu führen, dass wichtige Jobs hinter Jobs „hängen bleiben“, die zuvor eingereicht wurden. Indem Sie eine andere Planungsrichtlinie angeben, können Sie Rechenressourcen Ihren spezifischen Anforderungen entsprechend zuweisen.

> **Note**
>
> Wenn Sie die spezifische Reihenfolge planen möchten, in der Jobs ausgeführt werden, verwenden Sie den `dependsOn` Parameter in, SubmitJobum die Abhängigkeiten für jeden Job anzugeben.

Wenn Sie eine Planungsrichtlinie erstellen und diese an eine Auftragswarteschlange anhängen, ist Fair-Share-Scheduling aktiviert. Wenn die Auftragswarteschlange über eine Planungsrichtlinie verfügt, bestimmt die Planungsrichtlinie die Reihenfolge, in der Jobs ausgeführt werden. Weitere Informationen finden Sie unter Verwenden Sie Fair-Share-Scheduling-Richtlinien, um Share-Identifikatoren zuzuweisen.

Themen

- Verwenden Sie gemeinsame Identifikatoren, um Workloads zu identifizieren
- Verwenden Sie Fair-Share-Scheduling-Richtlinien, um Share-Identifikatoren zuzuweisen
- Verwenden Sie Fair-Share-Scheduling, um Jobs besser planen zu können
- Tutorial: Eine Planungsrichtlinie erstellen
- Referenz: Vorlage für Planungsrichtlinien

Verwenden Sie gemeinsame Identifikatoren, um Workloads zu identifizieren

Sie können Share-IDs verwenden, um Jobs zu taggen und zwischen Benutzern und Workloads zu unterscheiden. Der AWS Batch Scheduler verfolgt die Nutzung für jede gemeinsame Kennung mithilfe der *`(T * weightFactor)`* Formel, wobei die vCPU im Zeitverlauf angegeben *T*ist. Der Scheduler wählt anhand der Share-ID Jobs mit der geringsten Auslastung aus. Sie können eine Share-ID verwenden, ohne sie zu überschreiben.

> **Note**
>
> Freigabekennungen sind innerhalb einer Auftragswarteschlange eindeutig und werden nicht in allen Auftragswarteschlangen zusammengefasst.

Sie können die Fair-Share-Scheduling-Priorität festlegen, um die Reihenfolge zu konfigurieren, in der Jobs anhand einer gemeinsamen Kennung ausgeführt werden. Jobs mit einer höheren

Planungspriorität werden zuerst geplant. Wenn Sie keine Fair-Share-Scheduling-Richtlinie angeben, werden alle Jobs, die in die Job-Warteschlange gestellt werden, in der FIFO-Reihenfolge geplant. Wenn Sie einen Job einreichen, können Sie weder eine Share-ID noch eine Fair-Share-Scheduling-Priorität angeben.

> **Note**
>
> Angehängte Rechenressourcen werden allen Share-IDs gleichmäßig zugewiesen, sofern sie nicht ausdrücklich außer Kraft gesetzt werden.

Verwenden Sie Fair-Share-Scheduling-Richtlinien, um Share-Identifikatoren zuzuweisen

Mithilfe von Planungsrichtlinien können Sie konfigurieren, wie Rechenressourcen in einer Auftragswarteschlange Benutzern oder Workloads zugewiesen werden. Mithilfe von Fair-Share-Scheduling-Richtlinien können Sie Workloads oder Benutzern unterschiedliche Share-IDs zuweisen. AWS Batch weist jeder Share-ID einen Prozentsatz der gesamten Ressourcen zu, die während eines bestimmten Zeitraums verfügbar sind.

Der Fair-Share-Prozentsatz wird anhand der `shareDecaySeconds` Werte und berechnet. `shareDistribution` Sie können der Fair-Share-Analyse mehr Zeit hinzufügen, indem Sie der Richtlinie eine Share Decay Time zuweisen. Durch das Hinzufügen von Zeit wird der Zeit mehr Gewicht und dem definierten Gewicht weniger Gewicht zugewiesen. Sie können Rechenressourcen für Share-IDs reservieren, die nicht aktiv sind, indem Sie eine Rechenreservierung angeben. Weitere Informationen finden Sie unter SchedulingPolicyDetail.

Verwenden Sie Fair-Share-Scheduling, um Jobs besser planen zu können

Fair-Share-Scheduling bietet eine Reihe von Steuerelementen, mit denen Sie Jobs besser planen können.

> **Note**
>
> Weitere Informationen zur Planung von Richtlinienparametern finden Sie unter. SchedulingPolicyDetail

- Share Decay-Sekunden — Der Zeitraum (in Sekunden), den der AWS Batch Scheduler verwendet, um für jede Share-ID einen Fair-Share-Prozentsatz zu berechnen. Ein Wert von Null gibt an, dass nur die aktuelle Nutzung gemessen wird. Eine längere Abklingzeit verleiht der Zeit mehr Gewicht.

> **Note**
>
> Der Zeitraum für den Zerfall wird wie folgt berechnet: *`shareDecaySeconds + OrderMinutes`*wo *`OrderMinutes`* ist die Zeit in der Reihenfolge in Minuten.

- Compute-Reservierung — Verhindert, dass Jobs in einem einzigen Share-Identifier alle Ressourcen verbrauchen, die an die Auftragswarteschlange angehängt sind. Bei der reservierten Quote `ActiveFairShares` handelt es *`(computeReservation/100)^ActiveFairShares`* sich um die Anzahl der aktiven Freigabekennungen.

> **Note**
>
> Wenn eine Share-ID Jobs in einem`SUBMITTED`,, `PENDINGRUNNABLE`, `STARTING` oder `RUNNING` Bundesstaat hat, wird sie als aktive Share-ID betrachtet. Nach Ablauf der Gültigkeitsdauer gilt eine Share-ID als inaktiv.

- Gewichtungsfaktor — Der Gewichtungsfaktor für die Aktienkennung. Der Standardwert lautet 1. Ein niedrigerer Wert ermöglicht die Ausführung von Aufträgen aus dem Share Identifier oder verleiht dem Share Identifier zusätzliche Laufzeit. Jobs, die eine gemeinsame Kennung mit einem Gewichtungsfaktor von 0,125 (1/8) verwenden, werden beispielsweise achtmal so viele Rechenressourcen zugewiesen wie Jobs, die eine gemeinsame ID mit einem Gewichtungsfaktor von 1 verwenden.

> **Note**
>
> Sie müssen dieses Attribut nur definieren, wenn Sie den Standard-Gewichtungsfaktor 1 aktualisieren müssen.

Wenn die Job-Queue aktiv ist und Jobs verarbeitet, können Sie eine Liste der ersten 100 `RUNNABLE` Jobs über den Job-Warteschlangen-Snapshot überprüfen. Weitere Informationen finden Sie unter Status der Auftragswarteschlange anzeigen.

Tutorial: Eine Planungsrichtlinie erstellen

Bevor Sie eine Auftragswarteschlange mit einer Planungsrichtlinie erstellen können, müssen Sie eine Planungsrichtlinie erstellen. Wenn Sie eine Fair-Share-Scheduling-Richtlinie erstellen, verknüpfen Sie eine oder mehrere Share Identifier oder Share-Identifier-Präfixe mit Gewichtungen für die Warteschlange und weisen der Richtlinie optional einen Verfallzeitraum und eine Rechenreservierung zu.

Um eine Planungsrichtlinie zu erstellen

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie die zu verwendende Region in der Navigationsleiste aus.
3. Wählen Sie im Navigationsbereich Scheduling Policies und Create aus.
4. Geben Sie unter Name einen eindeutigen Namen für Ihre Planungsrichtlinie ein. Bis zu 128 Buchstaben (Groß- und Kleinbuchstaben), Ziffern, Bindestriche und Unterstriche sind zulässig.
5. (Optional) Geben Sie für Share Decay Seconds einen ganzzahligen Wert für die Share Decay-Zeit der Fair-Share-Scheduling-Richtlinie ein. Bei einer längeren Share Decay-Zeit werden bei der Planung von Jobs über einen längeren Zeitraum deutlich mehr Rechenressourcen beansprucht. Dadurch können Jobs, die eine Share-ID verwenden, vorübergehend mehr Rechenressourcen verbrauchen, als es die Gewichtung dieser Share-ID zulassen würde, wenn diese Share-ID in letzter Zeit keine Rechenressourcen verwendet hätte.
6. (Optional) Geben Sie für Compute-Reservierung einen ganzzahligen Wert für die Rechenreservierung der Fair-Share-Scheduling-Richtlinie ein. Bei der Compute-Reservierung werden einige V CPUs reserviert, die für Share-Identifier verwendet werden können, die derzeit nicht aktiv sind.

   Bei der reservierten Quote handelt ActiveFairShareses `(computeReservation/100)^ActiveFairShares` sich um die Anzahl der aktiven Aktienkennungen.

   Ein `computeReservation` Wert von 50 gibt beispielsweise an, dass 50% der maximal verfügbaren VCPU reserviert werden AWS Batch sollten, wenn es nur eine gemeinsame Kennung gibt, 25%, wenn zwei Freigabekennungen vorhanden sind, und 12,5%, wenn es drei Share-IDs gibt. Ein `computeReservation` Wert von 25 gibt an, dass 25% der maximal verfügbaren VCPU reserviert werden AWS Batch sollten, wenn es nur eine gemeinsame Kennung gibt, 6,25%, wenn es zwei Share-IDs gibt, und 1.56%, wenn es drei Share-IDs gibt.

7. Im Abschnitt Share-Attribute können Sie die Share-ID und die Gewichtung für jede Share-ID angeben, die mit der Fair-Share-Scheduling-Richtlinie verknüpft werden sollen.

   a. Wählen Sie Aktienkennung hinzufügen aus.

   b. Geben Sie für Share Identifier die Share-ID an. Wenn die Zeichenfolge mit '*' endet, wird daraus ein Share-Identifier-Präfix, das verwendet wird, um Share-IDs für Jobs abzugleichen. Alle Share-Identifier und Share-Identifier-Präfixe in einer Planungsrichtlinie müssen eindeutig sein und dürfen sich nicht überschneiden. Beispielsweise können Sie in derselben Fair-Share-Scheduling-Richtlinie nicht das Präfix „UserA*“ und die Share-ID „UserA1“ als Präfix verwenden.

   c. Geben Sie unter Gewichtungsfaktor das relative Gewicht für die Aktienkennung an. Der Standardwert lautet 1.0. Ein niedrigerer Wert hat eine höhere Priorität für Rechenressourcen. Wenn ein Präfix für die gemeinsame Kennung verwendet wird, gilt für Jobs mit einer gemeinsamen Kennung, die mit dem Präfix beginnen, der gleiche Gewichtungsfaktor. Dadurch wird der Gewichtungsfaktor für diese Jobs effektiv erhöht, wodurch ihre individuelle Priorität verringert wird, aber derselbe Gewichtungsfaktor für das gemeinsame Identifizierungspräfix beibehalten wird.

8. (Optional) Im Abschnitt „Tags“ können Sie den Schlüssel und den Wert für jedes Tag angeben, das der Planungsrichtlinie zugeordnet werden soll. Weitere Informationen finden Sie unter Kennzeichnen Sie Ihre AWS Batch Ressourcen.

9. Wählen Sie Senden, um den Vorgang abzuschließen und Ihre Planungsrichtlinie zu erstellen.

Referenz: Vorlage für Planungsrichtlinien

Eine leere Vorlage für Planungsrichtlinien ist unten dargestellt. Sie können diese Vorlage verwenden, um Ihre Planungsrichtlinie zu erstellen, die dann in einer Datei gespeichert und mit der AWS CLI `--cli-input-json` Option verwendet werden kann. Weitere Informationen zu diesen Parametern finden Sie CreateSchedulingPolicyin der AWS Batch API-Referenz.

> **Note**
>
> Mit dem folgenden AWS CLI Befehl können Sie eine Vorlage für eine Job-Warteschlange generieren.
>
> ```
> $ aws batch create-scheduling-policy --generate-cli-skeleton
> ```

```
{
    "name": "",
    "fairsharePolicy": {
        "shareDecaySeconds": 0,
        "computeReservation": 0,
        "shareDistribution": [
            {
                "shareIdentifier": "",
                "weightFactor": 0.0
            }
        ]
    },
    "tags": {
        "KeyName": ""
    }
}
```

Ressourcenbewusste Planung

AWS Batch plant Jobs auf der Grundlage der vCPU, GPU und der Speicherverfügbarkeit in der Compute Environment (CE), die der Job Queue (JQ) zugeordnet ist. Aber manchmal garantiert allein die Verfügbarkeit dieser CE-Ressourcen nicht, dass der Job erfolgreich ausgeführt wird, da er möglicherweise von anderen benötigten Ressourcen abhängig ist, sodass diese Jobs storniert oder beendet werden. Dies führt zu einer ineffizienten Nutzung von Rechenressourcen. Um dieses Problem zu lösen, kann die ressourcenorientierte Planung die Verfügbarkeit abhängiger Ressourcen überprüfen, die nicht von CE stammen, bevor der Job für die Ausführung auf einem CE geplant wird.

AWS Batch Mit der ressourcenorientierten Planung können Sie Jobs auf der Grundlage verbrauchbarer Ressourcen planen, die für die Ausführung Ihrer Jobs benötigt werden — Lizenztoken von Drittanbietern, Bandbreite für den Datenbankzugriff, die Notwendigkeit, Aufrufe an eine Drittanbieter-API zu drosseln usw. Sie geben die verbrauchbaren Ressourcen an, die für die Ausführung eines Jobs benötigt werden, und Batch berücksichtigt diese Ressourcenabhängigkeiten bei der Planung eines Jobs. Sie können vermeiden, manuelle Eingriffe vorzunehmen, um Jobausfälle und lange Wartezeiten aufgrund eines Mangels an verbrauchbaren Ressourcen zu vermeiden. Sie können die Unterauslastung der Rechenressourcen verringern, indem Sie nur die Jobs zuweisen, für die alle erforderlichen Ressourcen verfügbar sind.

Die ressourcenorientierte Planung ist sowohl für FIFO- als auch für Fair-Share-Planungsrichtlinien verfügbar und kann mit allen von Batch unterstützten Computerplattformen verwendet werden,

einschließlich EKS, ECS und Fargate. Es kann mit Array-Jobs, Multi-Node-Parallel-Jobs (MNP) und mit regulären Batch-Jobs verwendet werden.

Um die ressourcenorientierte Planung zu konfigurieren, geben Sie zunächst alle verbrauchbaren Ressourcen an, die für die Ausführung Ihrer Jobs benötigt werden, sowie die Gesamtanzahl, die für jede Ressource verfügbar ist. Anschließend geben Sie für jeden Auftrag, der eine verbrauchbare Ressource benötigt, den Namen und die erforderlichen Mengen der einzelnen benötigten Ressourcen an. Batch verfolgt, wie viele verbrauchbare Ressourcen für die Jobs in Ihren Jobwarteschlangen verfügbar sind, und stellt sicher, dass ein Job nur dann ausgeführt wird, wenn alle benötigten verbrauchbaren Ressourcen für die erfolgreiche Ausführung des Jobs verfügbar sind.

Themen

- Erstellen Sie verbrauchbare Ressourcen
- Geben Sie die Ressourcen an, die für die Ausführung eines Jobs benötigt werden
- Prüfen Sie, wie viele Ressourcen verwendet und verfügbar sind
- Aktualisieren Sie die Menge einer Ressource, während sie von Aufträgen verwendet wird
- Finden Sie die Jobs, für die eine bestimmte verbrauchbare Ressource erforderlich ist
- Löschen Sie eine verbrauchbare Ressource

Erstellen Sie verbrauchbare Ressourcen

Sie müssen zunächst die verbrauchbaren Ressourcen erstellen, die die Nicht-CE-Ressourcen darstellen, die bei der Ausführung eines Jobs verbraucht werden und nur in begrenzten Mengen verfügbar sind. Jede verbrauchbare Ressource hat eine:

- Ressourcenname (`consumableResourceName`), der auf Kontoebene eindeutig sein muss.
- (optional) Ressourcentyp (`resourceType`), der angibt, ob die Ressource für die Wiederverwendung verfügbar ist, nachdem ein Job abgeschlossen wurde. Dies kann einer der folgenden sein:
  - `REPLENISHABLE` (Standard)
  - `NON_REPLENISHABLE`
- Gesamtmenge (`totalQuantity`), die die Gesamtmenge der verfügbaren verbrauchbaren Ressource angibt.

Die maximale Anzahl verbrauchbarer Ressourcen pro Konto beträgt 50.000.

Konsole:

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole die Option Consumable Resources aus.
2. Wählen Sie „Verbrauchbare Ressource erstellen“.
3. Geben Sie einen eindeutigen Ressourcennamen und die gesamte Ressourcenmenge ein und wählen Sie aus, ob die Art der Ressource „Wiederauffüllbar“ oder „Nicht wiederauffüllbar“ ist.
4. Wählen Sie „Verbrauchbare Ressource erstellen“.

API:

Verwenden Sie die `CreateConsumableResource`API, um die gewünschten Ressourcen zu definieren.

Geben Sie die Ressourcen an, die für die Ausführung eines Jobs benötigt werden

Wenn Sie einen Job registrieren, können Sie den Namen einer oder mehrerer Ressourcen angeben, die Sie erstellt haben (`consumableResource`), und die Menge dieser Ressource, die jede Instanz des Jobs benötigt (`quantity`).

Batch verfolgt die verfügbaren Einheiten jeder Ressource zu einem bestimmten Zeitpunkt. Für jeden Job in der Auftragswarteschlange stellt der Batch-Scheduler sicher, dass Ihr Job nur ausgeführt wird, wenn die angegebenen Ressourcenabhängigkeiten verfügbar sind.

Wenn eine nutzbare Ressource für den Job nicht verfügbar ist, wenn der Job den Anfang der Warteschlange erreicht, wartet der Job im `RUNNABLE` Status, bis alle erforderlichen Ressourcen verfügbar sind oder das Zeitlimit für den Jobstatus erreicht ist (sieheStatus der Auftragswarteschlange anzeigen). Sobald Batch bestätigt hat, dass alle Ressourcen verfügbar sind, wechselt der Job in den `STARTING` Status und dann in`RUNNING`. Ressourcen werden gesperrt, sobald der Job zu oder verschoben wird, `STARTING` und wieder entsperrt, wenn der Job zu `SUCCEEDED` oder wechselt`FAILED`.

Sie können auch die Menge einer Ressource aktualisieren, die für einen bestimmten Job benötigt wird, wenn Sie den Job weiterleiten.

Konsole:

So geben Sie Ressourcen und deren benötigte Mengen an, wenn Sie einen Job definieren:

1. Definieren Sie einen Job mithilfe des Assistenten für Auftragsdefinitionen in der AWS Batch Konsole (Jobdefinitionen -> Erstellen).
2. Wählen Sie im Assistenten unter Schritt 4: Container konfigurieren unter Verbrauchbare Ressource den Namen einer erforderlichen Ressource aus der Liste aus. Geben Sie im Feld Angeforderter Wert die Menge dieser Ressource ein, die für eine Instanz dieses Auftrags benötigt wird, und wählen Sie dann Verbrauchbare Ressource hinzufügen aus.
3. Wiederholen Sie den vorherigen Schritt für alle verbrauchbaren Ressourcen, die für den Job benötigt werden. Sie können für jeden Job, den Sie definieren, bis zu 5 Ressourcen angeben.
4. Sie sehen eine Liste der verbrauchbaren Ressourcen, die Sie erstellt haben, nachdem Sie den Assistenten für die Auftragsdefinition abgeschlossen haben, aber bevor Sie Jobdefinition erstellen wählen.

Gehen Sie wie folgt vor, um die benötigte Menge an Ressourcen zu aktualisieren, wenn Sie einen Job einreichen:

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole Jobs und anschließend Neuen Job einreichen aus.
2. Geben Sie im Assistenten unter Schritt 2: Überschreibungen konfigurieren unter Überschreibungen verbrauchbarer Ressourcen einen neuen angeforderten Wert für jede verbrauchbare Ressource ein, deren benötigte Menge Sie für den Job überschreiben möchten.
3. Nachdem Sie alle Überschreibungen, die Sie für diesen Job vornehmen möchten, abgeschlossen haben, klicken Sie auf Weiter, um mit Prüfen und Absenden fortzufahren.

API:

Wenn Sie einen Job bei der `RegisterJobDefinition`API registrieren, geben Sie `consumableResourceList` im `consumableResourceProperties` Teil der Anfrage die verbrauchbaren Ressourcen an, die für die Ausführung einer Instanz des Jobs erforderlich sind, sowie deren Menge.

Wenn Sie einen Job mit der `SubmitJob`API einreichen, können Sie die Liste der verbrauchbaren Ressourcen und deren Menge überschreiben, indem Sie den `consumableResourcePropertiesOverride` Teil der Anfrage verwenden. Beachten Sie, dass

dadurch nur die Menge der Ressourcen außer Kraft gesetzt wird, die für jede Instanz des Jobs benötigt wird, nicht die verfügbare Gesamtmenge.

Prüfen Sie, wie viele Ressourcen verwendet und verfügbar sind

Mit Batch können Sie die Anzahl der verfügbaren Ressourcen (`availableQuantity`), die Anzahl der verwendeten Ressourcen (`inUseQuantity`) und die Gesamtressourcen (`totalQuantity`) zu einem bestimmten Zeitpunkt abfragen.

Sobald ein Auftrag den `STARTING` Status erreicht hat, werden die verbrauchten Ressourcen von der verfügbaren Menge dieser Ressource abgezogen. Wenn es sich bei der Ressource um eine Ressource handelt`REPLENISHABLE`, wird die Anzahl der verbrauchten Ressourcen wieder zur verfügbaren Menge hinzugefügt, sobald der Job entweder in den Status ERFOLGREICH ODER FEHLGESCHLAGEN übergegangen ist, und die Gesamtmenge bleibt unverändert. Ist dies der Fall`NON_REPLENISHABLE`, wird die Anzahl der verbrauchten Ressourcen sowohl von der Gesamtmenge als auch von der verfügbaren Menge subtrahiert und nicht wieder hinzugefügt, unabhängig davon, ob der Auftrag in den Status `SUCCEEDED` oder `FAILED` wechselt.

> **Note**
>
> Diese Information kann bis zu 30 Sekunden verzögert werden.

Konsole:

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole die Option Consumable Resources aus.
2. Wählen Sie entweder die Registerkarte Wiederauffüllbar oder Nicht wiederauffüllbar, um die Ressourcen dieses Typs anzuzeigen, die Sie erstellt haben.
3. Für jede auffüllbare Ressource zeigt die Konsole den Namen, die Gesamtmenge der Ressource, die Anzahl, die derzeit verwendet wird und wie viele noch verfügbar sind, zusammen mit einer Berechnung der Auslastung (die Anzahl der verwendeten Ressourcen geteilt durch die Gesamtmenge dieser Ressource) an.

   Für jede nicht auffüllbare Ressource zeigt die Konsole den Namen, die Anzahl der aktuell verwendeten Ressourcen und die Anzahl der Ressourcen an, die noch verfügbar sind.

Aktuelle Informationen zu verbrauchbaren Ressourcen können Sie auch auf einer Auftragsdetailseite in der Konsole einsehen.

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole Jobs und anschließend den Namen eines Jobs aus, um die Detailseite für diesen Job zu öffnen.
2. Informationen sowohl zu auffüllbaren als auch zu nicht auffüllbaren Ressourcen können angezeigt werden, sofern sie für den Job erforderlich sind. Für beide Typen zeigt die Konsole den Namen der Ressource, die angeforderte Menge für den Auftrag, die Anzahl der noch verfügbaren Ressourcen, die Zahl, die derzeit verwendet wird, die Gesamtmenge der Ressource sowie eine Berechnung der aktuellen Auslastung an (die Anzahl der Ressourcen, die pro Auftrag verwendet werden, geteilt durch die Gesamtmenge dieser Ressource).

API:

Verwenden Sie die `DescribeConsumableResourceAPI`, die die folgenden Informationen zurückgibt:

```
{
   "availableQuantity": number,
   "consumableResourceArn": "string",
   "consumableResourceName": "string",
   "createdAt": number,
   "inUseQuantity": number,
   "resourceType": "string",
   "tags": {
      "string" : "string"
   },
   "totalQuantity": number
}
```

Die `ListConsumableResourcesAPI` meldet auch die Anzahl der verwendeten Ressourcen (`inUseQuantity`) und die Gesamtzahl der derzeit verfügbaren Ressourcen (`totalQuantity`) als Teil ihrer Liste aller verbrauchbaren Ressourcen, die Sie in Ihrem Konto erstellt haben. Mit dieser API können Sie die Abfrage zur Liste der verbrauchbaren Ressourcen auch anhand des Namens der verbrauchbaren Ressourcen filtern.

Aktualisieren Sie die Menge einer Ressource, während sie von Aufträgen verwendet wird

Sie können die Gesamtmenge einer Ressource auf einen neuen Wert zurücksetzen, die Gesamtmenge erhöhen oder von ihr subtrahieren.

Wenn die neue Gesamtmenge, die Sie angeben, höher als zuvor ist, plant Batch weitere Jobs entsprechend ein. Wenn die neue Gesamtmenge geringer ist als zuvor und keine Einheiten dieser Ressource verwendet werden, reduziert Batch lediglich die Gesamtmenge (oder die verfügbare Menge). Wenn Einheiten verwendet werden, reduziert Batch die verfügbare Menge sofort, und wenn die Aufträge abgeschlossen sind, reduziert Batch die (verfügbare) Gesamtmenge, sodass letztendlich die neue Anzahl erreicht wird.

Konsole:

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole die Option Consumable Resources aus.
2. Wählen Sie entweder die Registerkarte Wiederauffüllbar oder Nicht wiederauffüllbar, um die Ressourcen dieses Typs anzuzeigen, die Sie erstellt haben.
3. Für wiederauffüllbare Ressourcen:

   1. Wählen Sie die Ressource aus, die Sie aktualisieren möchten, klicken Sie dann auf Aktionen und dann auf Ressourcen festlegen, Ressourcen hinzufügen oder Ressourcen entfernen.
   2. Es erscheint ein Popup-Fenster, in dem Sie Gesamtwert festlegen, Ressourcen hinzufügen oder Ressourcen entfernen können, je nachdem, welche Aktion Sie im vorherigen Schritt ausgewählt haben. Geben Sie die Menge ein, die Sie als neuen Gesamtwert festlegen möchten, die Sie zur Gesamtmenge hinzufügen oder die Sie von der Gesamtmenge subtrahieren möchten, und wählen Sie dann Ok aus.

   Für Ressourcen, die nicht wieder aufgefüllt werden können:

   1. Wählen Sie die Ressource aus, die Sie aktualisieren möchten, klicken Sie dann auf Aktionen und dann auf Ressourcen festlegen, Ressourcen hinzufügen oder Ressourcen entfernen.
   2. Es erscheint ein Popup-Fenster, in dem Sie Verfügbaren Wert festlegen, Ressourcen hinzufügen oder Ressourcen entfernen können, je nachdem, welche Aktion Sie im vorherigen Schritt ausgewählt haben. Geben Sie die Menge ein, die Sie als neuen verfügbaren

Wert festlegen möchten, die Sie zur verfügbaren Menge hinzufügen oder die Sie von der verfügbaren Menge subtrahieren möchten, und wählen Sie dann Ok aus.

API:

Verwenden Sie die `UpdateConsumableResourceAPI`, um eine neue Gesamtmenge für die Ressource festzulegen oder um die Gesamtmenge zu erhöhen oder zu reduzieren.

Finden Sie die Jobs, für die eine bestimmte verbrauchbare Ressource erforderlich ist

Mit Batch können Sie eine Liste von Jobs abrufen, für die eine bestimmte verbrauchbare Ressource erforderlich ist.

Konsole:

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole die Option Consumable Resources aus.
2. Wählen Sie in der Liste den Namen der verbrauchbaren Ressource aus. Die Detailseite für diese Ressource wird geöffnet.
3. Geben Sie unter Jobs suchen alle Filter ein, die Sie auf die Jobliste anwenden möchten. Sie können nach dem Jobnamen („entspricht“, „beginnt mit“), dem Datumsbereich (als der Job erstellt wurde) und nach zusätzlichen Kriterien („Jobwarteschlange“, „Jobdefinition“, „gemeinsame Job-ID“) filtern. Wählen Sie für jeden Filtertyp, den Sie anwenden möchten, eine der verfügbaren Optionen in der Drop-down-Liste aus und geben Sie alle erforderlichen zusätzlichen Informationen ein.

   Wählen Sie Search (Suchen) aus.
4. Eine (gefilterte) Liste von Aufträgen, für die die verbrauchbare Ressource erforderlich ist, wird angezeigt, einschließlich Name, Status, Anzahl der angeforderten Einheiten der verbrauchbaren Ressource, anderer benötigter verbrauchbarer Ressourcen usw. Mithilfe dieser Liste können Sie einen oder mehrere Jobs zum Stornieren oder Beenden auswählen. Sie können auch den Namen eines Jobs auswählen, um die Detailseite dieses Jobs zu öffnen.
5. Unter Jobs suchen können Sie jetzt Ergebnisse aktualisieren oder Suche löschen und von vorne beginnen.

API:

Mit der `ListJobsByConsumableResourceAPI` können Sie eine Liste von Jobs abrufen, die eine bestimmte verbrauchbare Ressource verwenden. Mit dieser API können Sie die Joblistenabfrage auch anhand des Jobstatus oder des Jobnamens filtern.

Löschen Sie eine verbrauchbare Ressource

Sie können eine verbrauchbare Ressource jederzeit löschen, auch wenn die Jobs, für die die Ressource erforderlich ist, noch ausgeführt werden. Sobald eine verbrauchbare Ressource gelöscht wurde, besteht möglicherweise eine Lücke zwischen dem Zeitpunkt, zu dem der Löschbefehl empfangen wird, und dem Job-Scheduler, der den Löschvorgang berücksichtigt. Es ist also möglich, dass Jobs, die die Ressource verbrauchen, direkt nach dem Löschaufruf geplant werden. Wenn die gelöschte verbrauchbare Ressource vom Ressourcentyp (`resourceType`) ist, wird dies ignoriert`REPLENISHABLE`, wenn die Jobs abgeschlossen sind. Wenn Sie eine verbrauchbare Ressource löschen und sie mit demselben Namen neu erstellen, wird sie als dieselbe Ressource betrachtet und kann von Aufträgen verwendet werden. `RUNNABLE`

Konsole:

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole die Option Consumable Resources aus.
2. Wählen Sie entweder die Registerkarte Wiederauffüllbar oder Nicht wiederauffüllbar, um die Ressourcen dieses Typs anzuzeigen, die Sie erstellt haben.
3. Wählen Sie jede Ressource aus, die Sie löschen möchten, und wählen Sie dann Löschen. Ein Popup-Fenster „Verbrauchbare Ressource löschen“ wird angezeigt. Um die Löschung zu bestätigen, klicken Sie auf Delete (Löschen).

Sie können eine verbrauchbare Ressource auch von der zugehörigen Detailseite in der Konsole löschen.

1. Wählen Sie im linken Navigationsbereich der AWS Batch Konsole die Option „Verbrauchbare Ressourcen“ aus.
2. Wählen Sie entweder die Registerkarte Wiederauffüllbar oder Nicht wiederauffüllbar, um die Ressourcen dieses Typs anzuzeigen, die Sie erstellt haben.
3. Wählen Sie den Namen der Ressource, die Sie löschen möchten. Die Detailseite der verbrauchbaren Ressource wird angezeigt. Wählen Sie Löschen aus. Ein Popup-Fenster „Verbrauchbare Ressource löschen“ wird angezeigt. Um die Löschung zu bestätigen, klicken Sie auf Delete (Löschen).

API:

Verwenden Sie die `DeleteConsumableResourceAPI`, um eine verbrauchbare Ressource zu löschen.

Berufsdefinitionen

AWS Batch Jobdefinitionen geben an, wie Jobs ausgeführt werden sollen. Während jeder Auftrag auf eine Auftragsdefinition verweisen muss, können viele der Parameter, die in der Auftragsdefinition angegeben sind, zur Laufzeit überschrieben werden.

Einige der Attribute in einer Auftragsdefinition sind folgende:

- Welches Docker-Image soll mit dem Container in Ihrem Job verwendet werden?
- Wie viele V CPUs und wie viel Speicher soll mit dem Container verwendet werden.
- Der Befehl, den der Container ausführen soll, wenn er gestartet wird.
- Welche (falls vorhanden) Umgebungsvariablen sollten beim Start an den Container übergeben werden.
- Alle Datenvolumes, die mit dem Container verwendet werden sollen.
- Welche (falls vorhanden) IAM-Rolle sollte Ihr Job für AWS Berechtigungen verwenden?

Inhalt

- Erstellen Sie eine Auftragsdefinition mit einem Knoten
- Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten
- Referenz: Jobdefinitionsvorlage, die verwendet ContainerProperties
- Erstellen Sie Jobdefinitionen mit EcsProperties
- Verwenden Sie den awslogs-Protokolltreiber
- Geben Sie sensible Daten an
- Authentifizierung in der privaten Registrierung für Jobs
- Amazon EFS-Volumes
- Referenz: Beispiele für Berufsdefinitionen

Erstellen Sie eine Auftragsdefinition mit einem Knoten

Bevor Sie Jobs in ausführen können AWS Batch, müssen Sie eine Jobdefinition erstellen. Dieser Prozess unterscheidet sich geringfügig zwischen parallel Aufträgen mit einem Knoten und mehreren Knoten. In diesem Thema wird speziell behandelt, wie Sie eine Jobdefinition für einen AWS Batch

Job erstellen, bei dem es sich nicht um einen parallel Job mit mehreren Knoten handelt (auch bekannt als Gruppenplanung).

Sie können eine parallel Jobdefinition mit mehreren Knoten auf Amazon Elastic Container Service-Ressourcen erstellen. Weitere Informationen finden Sie unter the section called “Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten”.

Themen

- Tutorial: Erstellen Sie eine Jobdefinition mit einem Knoten auf Amazon-Ressourcen EC2
- Tutorial: Erstellen Sie eine Einzelknoten-Jobdefinition auf Fargate-Ressourcen
- Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten auf Amazon EKS-Ressourcen
- Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten und mehreren Containern auf Amazon-Ressourcen EC2

Tutorial: Erstellen Sie eine Jobdefinition mit einem Knoten auf Amazon-Ressourcen EC2

Gehen Sie wie folgt vor, um eine Einzelknoten-Auftragsdefinition auf Amazon Elastic Compute Cloud (Amazon EC2) -Ressourcen zu erstellen.

So erstellen Sie eine neue Jobdefinition auf EC2 Amazon-Ressourcen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im linken Navigationsbereich Jobdefinitionen aus.
4. Wählen Sie Erstellen aus.
5. Wählen Sie als Orchestrierungstyp Amazon Elastic Compute Cloud (Amazon EC2).
6. Deaktivieren Sie für die EC2 Plattformkonfiguration die Option Parallelverarbeitung mit mehreren Knoten aktivieren.
7. Geben Sie unter Name einen eindeutigen Namen für Ihre Jobdefinition ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
8. (Optional) Geben Sie für das Ausführungstimeout den Timeout-Wert (in Sekunden) ein. Das Ausführungs-Timeout ist die Zeitspanne, bis ein unvollendeter Job beendet wird. Wenn ein Versuch die Timeoutdauer überschreitet, wird der Versuch gestoppt und erhält einen Status.

`FAILED` Weitere Informationen finden Sie unter Timeouts bei der Job. Der Mindestwert beträgt 60 Sekunden.

9. (Optional) Aktivieren Sie die Zeitplanungspriorität. Geben Sie einen Wert für die Planungspriorität zwischen 0 und 100 ein. Höhere Werte erhalten eine höhere Priorität.
10. (Optional) Geben Sie für Auftragsversuche ein, wie oft AWS Batch versucht wird, den Job in den `RUNNABLE` Status zu versetzen. Geben Sie eine Zahl zwischen 1 und 10 ein.
11. (Optional) Wählen Sie unter Bedingungen für die Wiederholungsstrategie die Option Beim Beenden bewerten hinzufügen aus. Geben Sie mindestens einen Parameterwert ein und wählen Sie dann eine Aktion aus. Für jeden Satz von Bedingungen muss Aktion entweder auf Erneut versuchen oder Beenden gesetzt sein. Diese Aktionen bedeuten Folgendes:
    - Wiederholen — AWS Batch Wiederholt die Versuche, bis die von Ihnen angegebene Anzahl von Auftragsversuchen erreicht ist.
    - Beenden — AWS Batch beendet die Wiederholung des Jobs.

    > **Important**
    > Wenn Sie „Beim Beenden bewerten hinzufügen“ wählen, müssen Sie mindestens einen Parameter konfigurieren und entweder eine Aktion auswählen oder „Beim Beenden auswerten entfernen“ wählen.

12. (Optional) Erweitern Sie „Tags“ und wählen Sie dann „Tag hinzufügen“, um der Ressource Tags hinzuzufügen. Geben Sie einen Schlüssel und einen optionalen Wert ein und wählen Sie dann Tag hinzufügen.
13. (Optional) Aktivieren Sie die Option Tags weitergeben, um Tags aus dem Auftrag und der Auftragsdefinition an die Amazon ECS-Aufgabe weiterzugeben.
14. Wählen Sie „Nächste Seite“.
15. Gehen Sie im Abschnitt Container-Konfiguration wie folgt vor:
    a. Wählen Sie unter Bild das Docker Bild aus, das Sie für Ihren Job verwenden möchten. Images in der Docker Hub-Registrierung sind standardmäßig verfügbar. Sie können auch andere Repositorys mit `repository-url/image:tag` angeben. Der Name kann bis zu 225 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-), Unterstriche (_), Doppelpunkte (:), Schrägstriche (/) und Zahlenzeichen (#) enthalten.

Dieser Parameter wird `Image` zugeordnet im Abschnitt Create a container (Erstellen eines Containers) im Docker-Remote-API und dem `IMAGE`-Parameter von docker run.

> **Note**
>
> Die Docker-Image-Architektur muss der Prozessorarchitektur der Compute-Ressourcen entsprechen, für die sie geplant sind. So können beispielsweise ARM-basierte Docker-Images nur auf ARM-basierten Compute-Ressourcen ausgeführt werden.

- Bilder in öffentlichen Amazon ECR-Repositorys verwenden die vollständigen `registry/repository[:tag]` oder die `registry/repository[@digest]` Namenskonventionen (z. B.`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Bilder in Amazon ECR-Repositorys verwenden die vollständige `registry/repository[:tag]` Namenskonvention (z. B. *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Images in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B `amazon/amazon-ecs-agent`).
- Image in anderen Online-Repositorys sind durch einen Domain-Namen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

b. Geben Sie für Command die Befehle als entsprechende JSON-Array-Zeichenfolge in das Feld ein.

   Dieser Parameter ist `Cmd` im Abschnitt Erstellen eines Containers der Docker Remote-API und dem Parameter `COMMAND` von docker run zugeordnet. Weitere Informationen zu dem Docker `CMD` Parameter finden Sie unter https://docs.docker.com/engine/reference/builder/#cmd.

   > **Note**
   >
   > Sie können Standardwerte für die Parameterersetzung und Platzhalter in Ihrem Befehl verwenden. Weitere Informationen finden Sie unter Parameter.

c. (Optional) Geben Sie für die Ausführungsrolle eine IAM-Rolle an, die den Amazon ECS-Container-Agenten die Erlaubnis erteilt, AWS API-Aufrufe in Ihrem Namen durchzuführen. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgaben. Weitere Informationen finden Sie unter IAM-Rollen für die Ausführung von Amazon ECS-Aufgaben im Amazon Elastic Container Service Developer Guide.

d. Wählen Sie für die Konfiguration der Job Role eine IAM-Rolle aus, die über Berechtigungen für die AWS APIs verfügt. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgaben. Weitere Informationen finden Sie unter IAM-Rollen für Aufgaben im Entwicklerhandbuch zum Amazon Elastic Container Service.

> **Note**
>
> Hier werden nur Rollen angezeigt, für die die Amazon Elastic Container Service Task Role Vertrauensstellung besteht. Weitere Informationen zum Erstellen einer IAM-Rolle für Ihre AWS Batch Jobs finden Sie unter Creating an IAM Role and Policy for your Tasks im Amazon Elastic Container Service Developer Guide.

16. Wählen Sie unter Parameter die Option Parameter hinzufügen, um Platzhalter für die Parametersetzung als Schlüssel - und optionale Wertepaare hinzuzufügen.

17. Gehen Sie im Abschnitt Umgebungskonfiguration wie folgt vor:

    a. Geben Sie für v CPUs die Zahl von v einCPUs , die für den Container reserviert werden sollen. Dieser Parameter ordnet zu `CpuShares` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--cpu-shares` für die docker run zu. Jede vCPU entspricht 1.024 CPU-Anteilen. Sie müssen mindestens eine vCPU angeben.

    b. Geben Sie unter Speicher das Speicherlimit ein, das für den Container verfügbar ist. Wenn Ihr Container versucht, die hier angegebene Speichermenge zu überschreiten, wird der Container gestoppt. Dieser Parameter ordnet zu `Memory` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--memory` für die docker run zu. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen.

> **Note**
>
> Um Ihre Ressourcennutzung zu maximieren, priorisieren Sie den Arbeitsspeicher für Jobs eines bestimmten Instance-Typs. Weitere Informationen finden Sie unter Speicherverwaltung für Rechenressourcen.

c. Wählen Sie unter Anzahl von die Anzahl der aus GPUs, die für den Container reserviert werden GPUs soll.

d. (Optional) Wählen Sie für Umgebungsvariablen die Option Umgebungsvariable hinzufügen aus, um Umgebungsvariablen als Name-Wert-Paare hinzuzufügen. Diese Variablen werden an den Container übergeben.

e. (Optional) Wählen Sie unter Secrets die Option Add secret aus, um Secrets als Name-Wert-Paare hinzuzufügen. Diese Geheimnisse werden im Container offengelegt. Weitere Informationen finden Sie unter:SecretOptionsLogConfiguration.

18. Wählen Sie Nächste Seite.

19. Gehen Sie im Abschnitt Linux-Konfiguration wie folgt vor:

    a. Für Benutzer geben Sie den Benutzernamen zur Verwendung innerhalb des Containers ein. Dieser Parameter ordnet zu `User` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--user` für die docker run zu.

    b. (Optional) Um dem Job-Container erweiterte Berechtigungen für die Host-Instance zu gewähren (ähnlich wie dem `root` Benutzer), ziehen Sie den Schieberegler Privileged nach rechts. Dieser Parameter ordnet zu `Privileged` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--privileged` für die docker run zu.

    c. (Optional) Aktivieren Sie die Option „Init-Prozess aktivieren“, um einen `init` Prozess innerhalb des Containers auszuführen. Dieser Prozess leitet Signale weiter und verarbeitet Prozesse.

20. (Optional) Gehen Sie im Abschnitt zur Dateisystemkonfiguration wie folgt vor:

    a. Aktivieren Sie die Option Schreibgeschütztes Dateisystem aktivieren, um den Schreibzugriff auf das Volume zu unterbinden.

    b. Geben Sie für Größe des gemeinsamen Speichers die Größe (in MiB) des `/dev/shm` Volumes ein.

    c. Geben Sie für Max. Swap-Größe die Gesamtmenge an Swap-Speicher (in MiB) ein, die der Container verwenden kann.

    d. Geben Sie für Swappiness einen Wert zwischen 0 und 100 ein, um das Swappiness-Verhalten des Containers anzugeben. Wenn Sie keinen Wert angeben und Swapping aktiviert ist, ist der Standardwert 60. Weitere Informationen finden Sie unter:Swappiness. LinuxParameters

    e. (Optional) Erweitern Sie Zusätzliche Konfiguration.

f. (Optional) Wählen Sie für Tmpfs Add tmpfs aus, um einen Mount hinzuzufügen. `tmpfs`

g. (Optional) Wählen Sie für Geräte die Option Gerät hinzufügen aus, um ein Gerät hinzuzufügen:

   i. Geben Sie für Container path (Container-Pfad) den Pfad in der Container-Instance an, um das der Host-Instance zugeordnete Gerät zugänglich zu machen. Wenn Sie dieses Feld leer lassen, wird der Hostpfad im Container verwendet.

   ii. Geben Sie für Host path (Host-Pfad) den Pfad eines Geräts in der Host-Instance an.

   iii. Wählen Sie unter Berechtigungen eine oder mehrere Berechtigungen aus, die auf das Gerät angewendet werden sollen. Die verfügbaren Berechtigungen sind READ, WRITE und MKNOD.

h. (Optional) Wählen Sie für die Volumes-Konfiguration die Option Volume hinzufügen aus, um eine Liste von Volumes zu erstellen, die an den Container übergeben werden sollen. Geben Sie den Namen und den Quellpfad für das Volume ein und wählen Sie dann Volume hinzufügen. Sie können sich auch dafür entscheiden, EFS aktivieren zu aktivieren.

i. (Optional) Wählen Sie für Bereitstellungspunkte die Option Konfiguration für Bereitstellungspunkte hinzufügen aus, um Bereitstellungspunkte für Datenvolumes hinzuzufügen. Sie müssen das Quellvolume und den Containerpfad angeben. Diese Einhängepunkte werden an die Instance Docker daemon auf einer Container-Instanz übergeben. Sie können sich auch dafür entscheiden, das Volume schreibgeschützt zu machen.

j. (Optional) Wählen Sie für die Ulimits-Konfiguration die Option Ulimit hinzufügen aus, um einen `ulimits` Wert für den Container hinzuzufügen. Geben Sie die Werte Name, Soft Limit und Hard Limit ein und wählen Sie dann Add ulimit aus.

21. Gehen Sie im Abschnitt Task-Eigenschaften wie folgt vor:

    a. Wählen Sie für Ausführungsrolle — bedingt eine Rolle aus, damit Amazon ECS-Agenten AWS API-Aufrufe in Ihrem Namen tätigen können. Weitere Informationen zur Erstellung einer Ausführungsrolle finden Sie unter Tutorial: Erstellen Sie die IAM-Ausführungsrolle.

    b. Wählen Sie den ECS-Ausführungsbefehl aktivieren, um den direkten Zugriff auf die Amazon ECS-Container-Shell zu aktivieren und das Host-Betriebssystem zu umgehen. Sie müssen eine Aufgabenrolle wählen.

> **Important**
>
> Der ECS-Befehl Execute setzt voraus, dass das Dateisystem beschreibbar ist.

c. Wählen Sie für Aufgabenrolle eine Amazon ECS Identity and Access Management (IAM) -Rolle aus, damit der Container AWS API-Aufrufe in Ihrem Namen tätigen kann. Weitere Informationen finden Sie unter Amazon ECS-Aufgaben-IAM-Rolle im Amazon Elastic Container Service Developer Guide.

22. (Optional) Gehen Sie im Abschnitt zur Konfiguration der Protokollierung wie folgt vor:

    a. Wählen Sie unter Protokolltreiber den Protokolltreiber aus, den Sie verwenden möchten. Weitere Informationen zu den verfügbaren Protokolltreibern finden Sie unter:LogDriverLogConfiguration.

    > **Note**
    >
    > Standardmäßig wird der `awslogs` Protokolltreiber verwendet.

    b. Wählen Sie unter Optionen die Option Hinzufügen aus, um eine Option hinzuzufügen. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Option hinzufügen.

    c. Wählen Sie für Geheimnisse die Option Geheimnis hinzufügen aus. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Geheimnis hinzufügen aus, um ein Geheimnis hinzuzufügen.

    > **Tip**
    >
    > Weitere Informationen finden Sie unter:Geheime Optionen. LogConfiguration

23. Wählen Sie Nächste Seite.

24. Lesen Sie zur Überprüfung der Jobdefinition die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Jobdefinition erstellen.

Tutorial: Erstellen Sie eine Einzelknoten-Jobdefinition auf Fargate-Ressourcen

Gehen Sie wie folgt vor, um eine Auftragsdefinition mit einem Knoten für Ressourcen zu erstellen. AWS Fargate

So erstellen Sie eine neue Jobdefinition auf Fargate Resources:

1. Öffnen Sie die AWS Batch Konsole unter. https://console.aws.amazon.com/batch/
2. Wählen Sie in der oberen Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im linken Navigationsbereich Jobdefinitionen aus.
4. Wählen Sie Erstellen aus.
5. Wählen Sie als Orchestrierungstyp Fargate aus. Weitere Informationen finden Sie unter Fargate-Computerumgebungen.
6. Geben Sie unter Name einen eindeutigen Namen für Ihre Jobdefinition ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
7. (Optional) Geben Sie für das Ausführungs-Timeout den Timeout-Wert (in Sekunden) ein. Das Ausführungstimeout ist die Zeitspanne, bis ein unvollendeter Job beendet wird. Wenn ein Versuch die Timeoutdauer überschreitet, wird der Versuch gestoppt und erhält einen Status. `FAILED` Weitere Informationen finden Sie unter Timeouts bei der Job. Der Mindestwert beträgt 60 Sekunden.
8. (Optional) Aktivieren Sie die Zeitplanungspriorität. Geben Sie einen Wert für die Planungspriorität zwischen 0 und 100 ein. Höheren Werten wird eine höhere Priorität gegenüber niedrigeren Werten eingeräumt.
9. (Optional) Erweitern Sie Tags und wählen Sie dann Tag hinzufügen aus, um der Ressource Tags hinzuzufügen. Aktivieren Sie die Option „Tags weitergeben“, um Tags aus dem Job und der Auftragsdefinition zu übertragen.
10. Gehen Sie im Abschnitt zur Konfiguration der Fargate-Plattform wie folgt vor:
    a. Wählen Sie für die Runtime-Plattform die Architektur der Rechenumgebung aus.
    b. Wählen Sie unter Betriebssystemfamilie das Betriebssystem für die Rechenumgebung aus.
    c. Wählen Sie für CPU-Architektur die vCPU-Architektur aus.
    d. Geben Sie `LATEST` für die Fargate-Plattformversion eine bestimmte Version der Laufzeitumgebung ein.

e. (Optional) Aktivieren Sie „Öffentliche IP zuweisen“, um einer Fargate-Job-Netzwerkschnittstelle eine öffentliche IP-Adresse zuzuweisen. Für einen Auftrag, der in einem privaten Subnetz ausgeführt wird, um ausgehenden Datenverkehr an das Internet zu senden, muss für das private Subnetz ein NAT-Gateway angeschlossen sein, um Anfragen an das Internet weiterzuleiten. Möglicherweise möchten Sie dies tun, damit Sie Container-Images abrufen können. Weitere Informationen finden Sie unter Vernetzung von Amazon-ECS-Aufgaben im Entwicklerhandbuch zum Amazon Elastic Container Service.

f. (Optional) Geben Sie für kurzlebigen Speicher die Menge an flüchtigem Speicher ein, der der Aufgabe zugewiesen werden soll. Die Menge an flüchtigem Speicher muss zwischen 21 GiB und 200 GiB liegen. Standardmäßig werden 20 GiB flüchtiger Speicher zugewiesen, wenn Sie keinen Wert eingeben.

> **Note**
>
> Für kurzlebigen Speicher ist die Fargate-Plattformversion 1.4 oder höher erforderlich.

g. Geben Sie für die Ausführungsrolle eine IAM-Rolle an, die dem Amazon ECS-Container und den Fargate-Agenten die Erlaubnis erteilt, AWS API-Aufrufe in Ihrem Namen durchzuführen. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen, einschließlich der Voraussetzungen für die Konfiguration, finden Sie unter IAM-Rollen für die Ausführung von Amazon ECS-Aufgaben im Amazon Elastic Container Service Developer Guide.

h. Geben Sie für Auftragsversuche ein, wie oft AWS Batch versucht wird, den Job in einen `RUNNABLE` Status zu versetzen. Geben Sie eine Zahl zwischen 1 und 10 ein.

i. Optional) Wählen Sie unter Bedingungen für die Wiederholungsstrategie die Option Beim Beenden bewerten hinzufügen aus. Geben Sie mindestens einen Parameterwert ein und wählen Sie dann eine Aktion aus. Für jeden Satz von Bedingungen muss Aktion entweder auf Wiederholen oder Beenden gesetzt werden. Diese Aktionen bedeuten Folgendes:

   - Wiederholen — AWS Batch Wiederholt die Versuche, bis die von Ihnen angegebene Anzahl von Auftragsversuchen erreicht ist.
   - Beenden — AWS Batch beendet die Wiederholung des Jobs.

> **Important**
>
> Wenn Sie „Beim Beenden bewerten hinzufügen“ wählen, müssen Sie mindestens einen Parameter konfigurieren und eine Aktion auswählen oder „Beim Beenden auswerten entfernen“ wählen.

11. Wählen Sie „Nächste Seite“.
12. Gehen Sie im Abschnitt Container-Konfiguration wie folgt vor:

    a. Wählen Sie unter Image das Docker-Image aus, das Sie für Ihren Job verwenden möchten. Images in der Docker-Hub-Registry sind standardmäßig verfügbar. Sie können auch andere Repositorys mit `repository-url/image:tag` angeben. Der Name kann bis zu 225 Zeichen lang sein. Sie kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-), Unterstriche (_), Doppelstriche (:), Punkte (.), Schrägstriche (/) und Zahlenzeichen (#) enthalten. Dieser Parameter wird `Image` zugeordnet im Abschnitt Create a container (Erstellen eines Containers) im Docker-Remote-API und dem `IMAGE`-Parameter von docker run.

    > **Note**
    >
    > Die Docker-Image-Architektur muss der Prozessorarchitektur der Compute-Ressourcen entsprechen, für die sie geplant sind. So können beispielsweise ARM-basierte Docker-Images nur auf ARM-basierten Compute-Ressourcen ausgeführt werden.

    - Bilder in öffentlichen Amazon ECR-Repositorys verwenden die vollständigen `registry/repository[:tag]` oder die `registry/repository[@digest]` Namenskonventionen (z. B.`public.ecr.aws/registry_alias/my-web-app:latest`).
    - Bilder in Amazon ECR-Repositorys verwenden die vollständige `registry/repository[:tag]` Namenskonvention (z. B. `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
    - Images in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).

- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B `amazon/amazon-ecs-agent`).
- Image in anderen Online-Repositorys sind durch einen Domain-Namen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

b. Geben Sie für Command die Befehle als entsprechende JSON-Array-Zeichenfolge in das Feld ein.

Dieser Parameter ist `Cmd` im Abschnitt Erstellen eines Containers der Docker Remote-API und dem Parameter `COMMAND` von docker run zugeordnet. Weitere Informationen zu dem Docker `CMD` Parameter finden Sie unter https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> Sie können Standardwerte für die Parameterersetzung und Platzhalter in Ihrem Befehl verwenden. Weitere Informationen finden Sie unter Parameter.

c. (Optional) Fügen Sie der Auftragsdefinition Parameter als Name-Wert-Zuordnungen hinzu, um die Standardwerte der Auftragsdefinition zu überschreiben. So fügen Sie einen Parameter hinzu:

- Wählen Sie für Parameter die Option Parameter hinzufügen aus, geben Sie ein Name-Wert-Paar ein und wählen Sie dann Parameter hinzufügen aus.

> **Important**
>
> Wenn Sie Parameter hinzufügen wählen, müssen Sie entweder mindestens einen Parameter konfigurieren oder Parameter entfernen wählen

d. Gehen Sie im Abschnitt Umgebungskonfiguration wie folgt vor:

i. Wählen Sie für die Konfiguration der Jobrolle eine IAM-Rolle aus, die über Berechtigungen für die AWS APIs verfügt. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen finden Sie unter IAM-Rollen für Aufgaben im Entwicklerhandbuch zum Amazon Elastic Container Service.

> **Note**
>
> Hier werden nur Rollen angezeigt, für die die Amazon Elastic Container Service Task Role Vertrauensstellung besteht. Weitere Informationen zum Erstellen einer IAM-Rolle für Ihre AWS Batch Jobs finden Sie unter Creating an IAM Role and Policy for your Tasks im Amazon Elastic Container Service Developer Guide.

ii. Geben Sie für v die Zahl von v einCPUs, die für CPUs den Container reserviert werden soll. Dieser Parameter ordnet zu `CpuShares` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--cpu-shares` für die docker run zu. Jede vCPU entspricht 1.024 CPU-Anteilen. Sie müssen mindestens eine vCPU angeben.

iii. Geben Sie für Speicher das Speicherlimit ein, das für den Container verfügbar ist. Wenn Ihr Container versucht, den hier angegebenen Speicher zu überschreiten, wird der Container gestoppt. Dieser Parameter ordnet zu `Memory` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--memory` für die docker run zu. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen.

    Wenn Sie GuardDuty Runtime Monitoring verwenden, entsteht ein geringer Speicheraufwand für den GuardDuty Security Agent. Daher muss das Speicherlimit die Größe des GuardDuty Security Agents beinhalten. Informationen zu den Speicherlimits des GuardDuty Security Agents finden Sie unter CPU- und Speicherlimits im GuardDuty Benutzerhandbuch. Informationen zu den bewährten Methoden finden Sie im Amazon ECS-Entwicklerhandbuch unter Wie behebe ich Fehler bei Speichermangel bei meinen Fargate-Aufgaben nach der Aktivierung von Runtime Monitoring?.

> **Note**
>
> Um Ihre Ressourcennutzung zu maximieren, sollten Sie Arbeitsspeicher für Jobs eines bestimmten Instance-Typs priorisieren. Weitere Informationen finden Sie unter Speicherverwaltung für Rechenressourcen.

e. (Optional) Wählen Sie für Umgebungsvariablen die Option Umgebungsvariable hinzufügen aus, um Umgebungsvariablen als Name-Wert-Paare hinzuzufügen. Diese Variablen werden an den Container übergeben.

f. (Optional) Wählen Sie unter Secrets die Option Add secret aus, um Secrets als Name-Wert-Paare hinzuzufügen. Diese Geheimnisse werden im Container offengelegt. Weitere Informationen finden Sie unter:SecretOptionsLogConfiguration.

g. Wählen Sie „Nächste Seite“.

13. (Optional) Gehen Sie im Abschnitt Linux-Konfiguration wie folgt vor:

a. Geben Sie unter Benutzer einen Benutzernamen ein, der im Container verwendet werden soll.

b. Aktivieren Sie die Option Init-Prozess aktivieren, um einen Init-Prozess innerhalb des Containers auszuführen. Dieser Prozess leitet Signale weiter und verarbeitet Prozesse.

c. Aktivieren Sie die Option Schreibgeschütztes Dateisystem aktivieren, um den Schreibzugriff auf das Volume zu unterbinden.

d. (Optional) Erweitern Sie Zusätzliche Konfiguration.

e. Wählen Sie unter Konfiguration der Bereitstellungspunkte die Option Konfiguration der Bereitstellungspunkte hinzufügen aus, um Bereitstellungspunkte für Datenvolumen hinzuzufügen. Sie müssen das Quellvolume und den Containerpfad angeben. Diese Einhängepunkte werden an die Instance Docker daemon auf einer Container-Instanz übergeben.

f. Wählen Sie für die Volumes-Konfiguration die Option Volume hinzufügen aus, um eine Liste von Volumes zu erstellen, die an den Container übergeben werden sollen. Geben Sie einen Namen und einen Quellpfad für das Volume ein und wählen Sie dann Volume hinzufügen.

g. Gehen Sie im Abschnitt Task-Eigenschaften wie folgt vor:

i. Wählen Sie für Ausführungsrolle — bedingt eine Rolle aus, damit Amazon ECS-Agenten AWS API-Aufrufe in Ihrem Namen tätigen können. Weitere Informationen zur Erstellung einer Ausführungsrolle finden Sie unterTutorial: Erstellen Sie die IAM-Ausführungsrolle.

ii. Wählen Sie den ECS-Ausführungsbefehl aktivieren, um den direkten Zugriff auf die Amazon ECS-Container-Shell zu aktivieren und das Host-Betriebssystem zu umgehen. Sie müssen eine Aufgabenrolle wählen.

> **Important**
>
> Der ECS-Befehl Execute setzt voraus, dass das Dateisystem beschreibbar ist.

iii. Wählen Sie für Aufgabenrolle eine Amazon ECS Identity and Access Management (IAM) -Rolle aus, damit der Container AWS API-Aufrufe in Ihrem Namen tätigen kann. Weitere Informationen finden Sie unter Amazon ECS-Aufgaben-IAM-Rolle im Amazon Elastic Container Service Developer Guide.

h. Im Abschnitt zur Konfiguration der Protokollierung:

i. (Optional) Wählen Sie unter Protokolltreiber den Protokolltreiber aus, den Sie verwenden möchten. Weitere Informationen zu den verfügbaren Protokolltreibern finden Sie unter:LogDriverLogConfiguration.

> **Note**
>
> Standardmäßig wird der `awslogs` Protokolltreiber verwendet.

ii. (Optional) Wählen Sie unter Optionen die Option Hinzufügen aus, um eine Option hinzuzufügen. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Option hinzufügen.

iii. (Optional) Wählen Sie unter Secrets die Option Secret hinzufügen aus, um ein Secret hinzuzufügen. Geben Sie dann ein Name-Wert-Paar ein und wählen Sie Geheimnis hinzufügen aus.

> **Tip**
>
> Weitere Informationen finden Sie unter:Geheime Optionen. LogConfiguration

14. Wählen Sie „Nächste Seite“.
15. Lesen Sie zur Überprüfung der Jobdefinition die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Jobdefinition erstellen.

Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten auf Amazon EKS-Ressourcen

Gehen Sie wie folgt vor, um eine Auftragsdefinition für einen einzelnen Knoten auf Amazon Elastic Kubernetes Service (Amazon EKS) zu erstellen.

So erstellen Sie eine neue Jobdefinition auf Amazon EKS-Ressourcen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der oberen Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im linken Navigationsbereich Jobdefinitionen aus.
4. Wählen Sie Erstellen aus.
5. Wählen Sie als Orchestrierungstyp Elastic Kubernetes Service (EKS) aus.
6. Geben Sie unter Name einen eindeutigen Namen für Ihre Jobdefinition ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
7. (Optional) Geben Sie für das Ausführungs-Timeout den Timeout-Wert (in Sekunden) ein. Das Ausführungs-Timeout ist die Zeitspanne, bis ein unvollendeter Job beendet wird. Wenn ein Versuch die Timeoutdauer überschreitet, wird der Versuch gestoppt und erhält einen Status. `FAILED` Weitere Informationen finden Sie unter Timeouts bei der Job. Der Mindestwert beträgt 60 Sekunden.
8. (Optional) Aktivieren Sie die Zeitplanungspriorität. Geben Sie einen Wert für die Planungspriorität zwischen 0 und 100 ein. Höheren Werten wird eine höhere Priorität gegenüber niedrigeren Werten eingeräumt.
9. (Optional) Erweitern Sie Tags und wählen Sie dann Tag hinzufügen aus, um der Ressource Tags hinzuzufügen.
10. Wählen Sie „Nächste Seite“.
11. Gehen Sie im Bereich podEKS-Eigenschaften wie folgt vor:
    a. Geben Sie als Dienstkontoname ein Konto ein, das eine Identität für Prozesse bereitstellt, die in einem ausgeführt pod werden.
    b. Schalten Sie das Host-Netzwerk ein, um das Kubernetes pod Netzwerkmodell zu verwenden, und öffnen Sie einen Listening-Port für eingehende Verbindungen. Schalten Sie diese Einstellung nur für ausgehende Kommunikation aus.
    c. Wählen Sie für die DNS-Richtlinie eine der folgenden Optionen aus:
        - Kein Wert (Null) — Der pod ignoriert die DNS-Einstellungen aus der Kubernetes Umgebung.
        - Standard — Der pod erbt die Konfiguration für die Namensauflösung von dem Knoten, auf dem er ausgeführt wird.

> **Note**
>
> Wenn keine DNS-Richtlinie angegeben ist, ist Standard nicht die Standard-DNS-Richtlinie. Stattdessen ClusterFirstwird verwendet.

- ClusterFirst— Jede DNS-Abfrage, die nicht mit dem konfigurierten Cluster-Domänensuffix übereinstimmt, wird an den Upstream-Nameserver weitergeleitet, der vom Knoten geerbt wurde.
- ClusterFirstWithHostNet— Wird verwendet, wenn das Host-Netzwerk aktiviert ist.

d. (Optional) Wählen Sie für Volumes die Option Volume hinzufügen aus und gehen Sie dann wie folgt vor:

   i. Fügen Sie einen Namen für Ihr Volume hinzu.

   ii. (Optional) Fügen Sie den Hostpfad für das Verzeichnis auf dem Host hinzu.

   iii. (Optional) Fügen Sie ein Medium und eine Größenbeschränkung hinzu, um ein Kubernetes EmptyDir zu konfigurieren.

   iv. (Optional) Geben Sie einen geheimen Namen für den Pod an und geben Sie an, ob das Geheimnis optional ist.

   v. (Optional) Definieren Sie einen Claim-Namen, um einen Kubernetes Persistent Volume Claim an den Pod anzuhängen, und legen Sie fest, ob er schreibgeschützt ist.

e. (Optional) Wählen Sie für Pod-Labels die Option Pod-Labels hinzufügen aus und geben Sie dann ein Name-Wert-Paar ein.

> **Important**
>
> Das Präfix für ein Pod-Label darf nicht `kubernetes.io/k8s.io/`, oder enthalten. `batch.amazonaws.com/`

f. (Optional) Wählen Sie für Pod-Anmerkungen die Option Anmerkungen hinzufügen aus und geben Sie dann ein Name-Wert-Paar ein.

> **Important**
>
> Das Präfix für eine Pod-Anmerkung darf nicht, oder enthaltenkubernetes.io/. `k8s.io/` `batch.amazonaws.com/`

g. Wählen Sie „Nächste Seite“.

h. Gehen Sie im Abschnitt Container-Konfiguration wie folgt vor:

   i. Geben Sie unter Name einen eindeutigen Namen für den Container ein. Der Name muss mit einem Buchstaben oder einer Zahl beginnen und kann bis zu 63 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen und Bindestriche (-) enthalten.

   ii. Wählen Sie unter Bild das Bild aus, das Sie für Docker Ihren Job verwenden möchten. Images in der Docker-Hub-Registry sind standardmäßig verfügbar. Sie können auch andere Repositorys mit `repository-url/image:tag` angeben. Er kann bis zu 255 Zeichen lang sein. Sie kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-), Unterstriche (_), Doppelstriche (:), Punkte (.), Schrägstriche (/) und Zahlenzeichen (#) enthalten. Dieser Parameter entspricht `Image` dem Abschnitt Create a container der Docker Remote API und dem `IMAGE` Parameter von docker run

> **Note**
>
> Die Docker-Image-Architektur muss der Prozessorarchitektur der Compute-Ressourcen entsprechen, für die sie geplant sind. So können beispielsweise ARM-basierte Docker-Images nur auf ARM-basierten Compute-Ressourcen ausgeführt werden.

- Bilder in öffentlichen Amazon ECR-Repositorys verwenden die vollständigen `registry/repository[:tag]` oder die `registry/repository[@digest]` Namenskonventionen (z. B.`public.ecr.aws/registry_alias/my-web-app:latest`).
- Bilder in Amazon ECR-Repositorys verwenden die vollständige `registry/repository[:tag]` Namenskonvention (z. B. `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
- Images in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B `amazon/amazon-ecs-agent`).
- Image in anderen Online-Repositorys sind durch einen Domain-Namen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

iii. (Optional) Wählen Sie für die Richtlinie zum Abrufen von Bildern aus, wann Bilder abgerufen werden sollen.

iv. (Optional) Geben Sie unter Command einen JSON Befehl ein, der an den Container übergeben werden soll.

v. (Optional) Geben Sie unter Argumente Argumente ein, die an den Container übergeben werden sollen. Wenn kein Argument angegeben wird, wird der Befehl container image verwendet.

i. (Optional) Sie können der Auftragsdefinition Parameter als Name-Wert-Zuordnungen hinzufügen, um die Standardeinstellungen der Auftragsdefinition zu überschreiben. So fügen Sie einen Parameter hinzu:

- Geben Sie für Parameter ein Name-Wert-Paar ein und wählen Sie dann Parameter hinzufügen aus.

> **Important**
>
> Wenn Sie Parameter hinzufügen wählen, müssen Sie mindestens einen Parameter konfigurieren oder Parameter entfernen wählen

j. Gehen Sie im Abschnitt Umgebungskonfiguration wie folgt vor:

i. Geben Sie für v CPUs die Zahl von v einCPUs , die für den Container reserviert werden sollen. Dieser Parameter ordnet zu `CpuShares` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--cpu-shares` für die docker run zu. Jede vCPU entspricht 1.024 CPU-Anteilen. Sie müssen mindestens eine vCPU angeben.

ii. Geben Sie unter Speicher das Speicherlimit ein, das für den Container verfügbar ist. Wenn Ihr Container versucht, den hier angegebenen Speicher zu überschreiten, wird der Container gestoppt. Dieser Parameter ordnet zu `Memory` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--memory` für die docker run zu. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen.

> **Note**
>
> Um Ihre Ressourcennutzung zu maximieren, priorisieren Sie den Arbeitsspeicher für Jobs eines bestimmten Instance-Typs. Weitere Informationen finden Sie unter Speicherverwaltung für Rechenressourcen.

k. (Optional) Wählen Sie für Umgebungsvariablen die Option Umgebungsvariable hinzufügen aus, um Umgebungsvariablen als Name-Wert-Paare hinzuzufügen. Diese Variablen werden an den Container übergeben.

l. (Optional) Für die Volume-Bereitstellung:

   i. Wählen Sie „Volume-Mount hinzufügen“.

   ii. Geben Sie einen Namen und anschließend einen Bereitstellungspfad in dem Container ein, in dem das Volume bereitgestellt ist. Geben Sie a ein SubPath, um anstelle des Stammverzeichnisses einen Unterpfad innerhalb des referenzierten Volumes anzugeben.

   iii. Wählen Sie Schreibgeschützt, um die Schreibberechtigungen für das Volume zu entfernen.

   iv. Wählen Sie „Volume-Mount hinzufügen“.

m. (Optional) Geben Sie unter Als Benutzer ausführen eine Benutzer-ID ein, um den Container-Prozess auszuführen.

> **Note**
>
> Die Benutzer-ID muss im Image vorhanden sein, damit der Container ausgeführt werden kann.

n. (Optional) Geben Sie für Als Gruppe ausführen eine Gruppen-ID ein, um die Laufzeit des Container-Prozesses auszuführen.

> **Note**
>
> Die Gruppen-ID muss im Image vorhanden sein, damit der Container ausgeführt werden kann.

o. (Optional) Um dem Container Ihres Jobs erhöhte Berechtigungen auf der Host-Instance zu gewähren (ähnlich wie dem `root` Benutzer), ziehen Sie den Schieberegler Privilegiert nach rechts. Dieser Parameter ordnet zu `Privileged` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--privileged` für die docker run zu.

p. (Optional) Aktivieren Sie das schreibgeschützte Root-Dateisystem, um den Schreibzugriff auf das Root-Dateisystem zu entfernen.

q. (Optional) Aktivieren Sie „Als Nicht-Root-Benutzer ausführen“, um die Container im als Nicht-Root-Benutzer auszuführen. pod

> **Note**
>
> Wenn „Als Nicht-Root ausführen“ aktiviert ist, kubelet validiert das das Image zur Laufzeit, um sicherzustellen, dass das Image nicht mit UID 0 ausgeführt wird.

r. Wählen Sie „Nächste Seite“.

12. Lesen Sie zur Überprüfung der Jobdefinition die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Jobdefinition erstellen.

Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten und mehreren Containern auf Amazon-Ressourcen EC2

Gehen Sie wie folgt vor, um eine Auftragsdefinition für einen einzelnen Knoten mit mehreren Containern auf Amazon Elastic Compute Cloud (Amazon EC2) -Ressourcen zu erstellen.

So erstellen Sie eine neue Jobdefinition auf EC2 Amazon-Ressourcen:

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im linken Navigationsbereich Jobdefinitionen aus.
4. Wählen Sie Erstellen aus.
5. Wählen Sie als Orchestrierungstyp Amazon Elastic Compute Cloud (Amazon EC2).
6. Deaktivieren Sie für Jobdefinitionsstruktur die Option Legacy-ContainerProperties-Strukturverarbeitung verwenden.
7. Deaktivieren Sie für die EC2 Plattformkonfiguration die Option Parallelverarbeitung mit mehreren Knoten aktivieren.
8. Wählen Sie Weiter aus.
9. Geben Sie im Abschnitt Allgemeine Konfiguration Folgendes ein:

a. Geben Sie unter Name einen eindeutigen Namen für Ihre Jobdefinition ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

b. Geben Sie für Ausführungstimeout — optional den Timeout-Wert (in Sekunden) ein. Das Ausführungs-Timeout ist die Zeitspanne, bis ein unvollendeter Job beendet wird. Wenn ein Versuch die Timeoutdauer überschreitet, wird der Versuch gestoppt und erhält einen Status. `FAILED` Weitere Informationen finden Sie unter Timeouts bei der Job. Der Mindestwert beträgt 60 Sekunden.

c. Aktivieren Sie die Priorität der Zeitplanung — optional. Geben Sie einen Wert für die Planungspriorität zwischen 0 und 100 ein. Höhere Werte erhalten eine höhere Priorität.

d. Erweitern Sie Tags — optional und wählen Sie dann Tag hinzufügen aus, um der Ressource Tags hinzuzufügen. Geben Sie einen Schlüssel und einen optionalen Wert ein und wählen Sie dann Tag hinzufügen.

e. Aktivieren Sie die Option Tags weitergeben, um Tags aus dem Job und der Auftragsdefinition an die Amazon ECS-Aufgabe weiterzugeben.

10. Geben Sie im Abschnitt Strategie wiederholen — optional Folgendes ein:

a. Geben Sie für Auftragsversuche ein, wie oft AWS Batch versucht wird, den Job in den `RUNNABLE` Status zu versetzen. Geben Sie eine Zahl zwischen 1 und 10 ein.

b. Wählen Sie unter Bedingungen für die Wiederholungsstrategie die Option Beim Beenden bewerten hinzufügen aus. Geben Sie mindestens einen Parameterwert ein und wählen Sie dann eine Aktion aus. Für jeden Satz von Bedingungen muss Aktion entweder auf Erneut versuchen oder Beenden gesetzt sein. Diese Aktionen bedeuten Folgendes:

- Wiederholen — AWS Batch Wiederholt die Versuche, bis die von Ihnen angegebene Anzahl von Auftragsversuchen erreicht ist.
- Beenden — AWS Batch beendet die Wiederholung des Auftrags.

> **Important**
>
> Wenn Sie „Beim Beenden bewerten hinzufügen“ wählen, müssen Sie mindestens einen Parameter konfigurieren und entweder eine Aktion auswählen oder „Beim Beenden auswerten entfernen“ wählen.

11. Geben Sie im Abschnitt Aufgabeneigenschaften Folgendes ein:

a. Wählen Sie für Ausführungsrolle — bedingt eine Rolle aus, damit Amazon ECS-Agenten AWS API-Aufrufe in Ihrem Namen tätigen können. Weitere Informationen zur Erstellung einer Ausführungsrolle finden Sie unterTutorial: Erstellen Sie die IAM-Ausführungsrolle.

b. Wählen Sie den ECS-Ausführungsbefehl aktivieren, um den direkten Zugriff auf die Amazon ECS-Container-Shell zu aktivieren und das Host-Betriebssystem zu umgehen. Sie müssen eine Aufgabenrolle wählen.

> **Important**
>
> Der ECS-Befehl Execute setzt voraus, dass das Dateisystem beschreibbar ist.

c. Wählen Sie für Aufgabenrolle eine Amazon ECS Identity and Access Management (IAM) -Rolle aus, damit der Container AWS API-Aufrufe in Ihrem Namen tätigen kann. Weitere Informationen finden Sie unter Amazon ECS-Aufgaben-IAM-Rolle im Amazon Elastic Container Service Developer Guide.

d. Wählen Sie für den IPC-Modus`host`,`task`, oder. `none` Wenn angegeben, teilen `host` sich alle Container, die sich innerhalb der Aufgaben befinden, die den Host-IPC-Modus auf derselben Container-Instance spezifiziert haben, dieselben IPC-Ressourcen mit der EC2 Host-Amazon-Instance. Wenn eine Aufgabe angegeben ist, teilen sich alle Container, die sich innerhalb der angegebenen Aufgabe befinden, dieselben IPC-Ressourcen. Wenn keine angegeben ist, sind IPC-Ressourcen in den Containern einer Aufgabe privat und werden nicht mit anderen Containern in einer Aufgabe oder auf der Container-Instance gemeinsam genutzt. Wenn kein Wert angegeben ist, hängt die gemeinsame Nutzung des IPC-Ressourcen-Namespace von der Einstellung des Docker-Daemons in der Container-Instance ab.

e. Wählen Sie für den PID-Modus `host` oder`task`. Für die Überwachung von Beiwagen benötigt `pidMode` möglicherweise Informationen über andere Container, die in der gleichen Aufgabe laufen. Wenn angegeben, teilen `host` sich alle Container innerhalb der Aufgaben, die den Host-PID-Modus auf derselben Container-Instance angegeben haben, denselben Prozess-Namespace mit der EC2 Host-Amazon-Instance. Wenn `task` angegeben ist, teilen sich alle Container innerhalb der angegebenen Aufgabe den gleichen Prozess-Namespace. Wenn kein Wert angegeben wird, ist der Standard ein privater Namespace für jeden Container.

12. Geben Sie im Abschnitt „Verbrauchbare Ressource“ Folgendes ein:

    a. Geben Sie einen eindeutigen Namen und den angeforderten Wert ein.

b. Sie können weitere verbrauchbare Ressourcen hinzufügen, indem Sie „Verbrauchbare Ressource hinzufügen“ wählen.

13. Geben Sie im Bereich Speicher Folgendes ein:

    a. Geben Sie einen Namen und einen Quellpfad für das Volume ein und wählen Sie dann Volume hinzufügen. Sie können sich auch dafür entscheiden, EFS aktivieren zu aktivieren.

    b. Sie können weitere Volumes hinzufügen, indem Sie Volume hinzufügen wählen.

14. Wählen Sie unter Parameter die Option Parameter hinzufügen aus, um Platzhalter für die Parametersetzung als Schlüssel - und optionale Wertepaare hinzuzufügen.

15. Wählen Sie „Nächste Seite“.

16. Gehen Sie im Abschnitt Container-Konfiguration wie folgt vor:

    a. Geben Sie unter Name einen Namen für den Container ein.

    b. Aktivieren Sie für Essential-Container die Option, wenn der Container essenziell ist.

    c. Wählen Sie unter Bild das Docker Bild aus, das Sie für Ihren Job verwenden möchten. Images in der Docker Hub-Registrierung sind standardmäßig verfügbar. Sie können auch andere Repositorys mit `repository-url/image:tag` angeben. Der Name kann bis zu 225 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-), Unterstriche (_), Doppelpunkte (:), Schrägstriche (/) und Zahlenzeichen (#) enthalten. Dieser Parameter wird `Image` zugeordnet im Abschnitt Create a container (Erstellen eines Containers) im Docker-Remote-API und dem `IMAGE`-Parameter von docker run.

    > **Note**
    >
    > Die Docker-Image-Architektur muss der Prozessorarchitektur der Compute-Ressourcen entsprechen, für die sie geplant sind. So können beispielsweise ARM-basierte Docker-Images nur auf ARM-basierten Compute-Ressourcen ausgeführt werden.

    - Bilder in öffentlichen Amazon ECR-Repositorys verwenden die vollständigen `registry/repository[:tag]` oder die `registry/repository[@digest]` Namenskonventionen (z. B.`public.ecr.aws/registry_alias/my-web-app:latest`).

- Bilder in Amazon ECR-Repositorys verwenden die vollständige `registry/repository[:tag]` Namenskonvention (z. B. *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*).
- Images in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B `amazon/amazon-ecs-agent`).
- Image in anderen Online-Repositorys sind durch einen Domain-Namen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

d. Konfigurieren Sie für die Ressourcenanforderungen jedes der folgenden Elemente:

   i. Wählen Sie für v CPUs die Nummer von CPUs für den Container.

   ii. Wählen Sie unter Speicher die Speichermenge für den Container aus.

   iii. Wählen Sie für GPU — optional die Nummer von GPUs für den Container aus.

e. Für Benutzer geben Sie den Benutzernamen zur Verwendung innerhalb des Containers ein.

f. Aktivieren Sie die Option „Schreibgeschütztes Dateisystem aktivieren“, um den Schreibzugriff auf das Volume zu unterbinden.

g. Aktivieren Sie „Privilegiert“, um dem Job-Container ähnlich wie dem Root-Benutzer erweiterte Rechte auf der Host-Instance zu gewähren.

h. Geben Sie für Command die Befehle als entsprechende JSON-String-Array-Entsprechung in das Feld ein.

   Dieser Parameter ist `Cmd` im Abschnitt Erstellen eines Containers der Docker Remote-API und dem Parameter `COMMAND` von docker run zugeordnet. Weitere Informationen zu dem Docker `CMD` Parameter finden Sie unter https://docs.docker.com/engine/reference/builder/#cmd.

   > **Note**
   >
   > Sie können Standardwerte für die Parameterersetzung und Platzhalter in Ihrem Befehl verwenden. Weitere Informationen finden Sie unter Parameter.

i. Geben Sie für Repository-Anmeldeinformationen — optional den ARN des Geheimnisses ein, das Ihre Anmeldeinformationen enthält.

j. Wählen Sie für Umgebungsvariablen — optional die Option Umgebungsvariablen hinzufügen aus, um Umgebungsvariablen hinzuzufügen, die an den Container übergeben werden sollen.

k. Gehen Sie im Abschnitt Linux-Parameter — optional wie folgt vor:

   i. Aktivieren Sie die Option Init-Prozess aktivieren, um einen Init-Prozess innerhalb des Containers auszuführen.

   ii. Geben Sie für Größe des gemeinsamen Speichers die Größe (in MiB) des Volumes ein the /dev/shm

   iii. Geben Sie für Max. Swap-Größe die Gesamtmenge an Swap-Speicher (in MiB) ein, die der Container verwenden kann.

   iv. Geben Sie für Swappiness einen Wert zwischen 0 und 100 ein, um das Swappiness-Verhalten des Containers anzugeben. Wenn Sie keinen Wert angeben und Swapping aktiviert ist, ist der Standardwert 60.

   v. Wählen Sie für Geräte die Option Gerät hinzufügen aus, um ein Gerät hinzuzufügen:

      A. Geben Sie für Container path (Container-Pfad) den Pfad in der Container-Instance an, um das der Host-Instance zugeordnete Gerät zugänglich zu machen. Wenn Sie dieses Feld leer lassen, wird der Hostpfad im Container verwendet.

      B. Geben Sie für Host path (Host-Pfad) den Pfad eines Geräts in der Host-Instance an.

      C. Wählen Sie unter Berechtigungen eine oder mehrere Berechtigungen aus, die auf das Gerät angewendet werden sollen. Die verfügbaren Berechtigungen sind READ, WRITE und MKNOD.

   vi. Wählen Sie für Tmpfs Add tmpfs aus, um einen Mount hinzuzufügen. `tmpfs`

l.

> **Note**
>
> Die Protokollierung von Firelens muss in einem speziellen Container erfolgen. Um die Firelens-Protokollierung zu konfigurieren:
>
> - Stellen Sie in jedem Container, mit Ausnahme Ihres dedizierten Firelens-Containers, den Logging-Treiber auf `awsfirelens`
> - Stellen Sie in Ihrem Firelens-Container die Firelens-Konfiguration — optional und die Logging-Konfiguration — optional auf das Logging-Ziel ein

Gehen Sie im Abschnitt Firelens-Konfiguration — optional wie folgt vor:

> **Important**
>
> AWS Batch erzwingt den `host` Netzwerkmodus für Amazon ECS-Jobs, die nicht MNP oder Fargate sind. Für Amazon ECS Firelens ist ein Root-Benutzer erforderlich. Bei der Ausführung von Aufgaben, die den `host` Netzwerkmodus verwenden, rät Amazon ECS aus Sicherheitsgründen davon ab, Container mit dem Root-Benutzer (UID 0) auszuführen. Daher entsprechen alle ECS-Jobs, die nicht MNP oder Fargate sind, mit Firelens-Protokollierung nicht den bewährten Sicherheitsmethoden.

i. Wählen Sie als Typ entweder oder. `fluentd` `fluentbit`

ii. Geben Sie unter Optionen das Name/Wert-Paar der Option ein. Mit der Option „Hinzugefügt“ können Sie weitere Optionen hinzufügen.

m. Gehen Sie im Abschnitt Konfiguration der Protokollierung — optional wie folgt vor:

i. Wählen Sie unter Protokolltreiber den Protokolltreiber aus, den Sie verwenden möchten. Weitere Informationen zu den verfügbaren Protokolltreibern finden Sie unter:LogDriverLogConfiguration.

> **Note**
>
> Standardmäßig wird der `awslogs` Protokolltreiber verwendet.

ii. Wählen Sie unter Optionen die Option Hinzufügen aus, um eine Option hinzuzufügen. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Option hinzufügen.

iii. Wählen Sie für Geheimnisse die Option Geheimnis hinzufügen aus. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Geheimnis hinzufügen aus, um ein Geheimnis hinzuzufügen.

> **Tip**
>
> Weitere Informationen finden Sie unter:Geheime Optionen. LogConfiguration

n. Wählen Sie unter Bereitstellungspunkte — optional die Option Bereitstellungspunkte hinzufügen aus, um Bereitstellungspunkte für Datenvolumes hinzuzufügen. Sie müssen das Quellvolume und den Containerpfad angeben.

o. Wählen Sie für Secrets — optional die Option Add secret aus, um ein Secret hinzuzufügen. Geben Sie dann ein Name-Wert-Paar ein und wählen Sie Geheimnis hinzufügen aus.

> **Tip**
>
> Weitere Informationen finden Sie unter:Geheime Optionen. LogConfiguration

p. Wählen Sie für Ulimits — optional die Option Add ulimit aus, um einen Wert für den Container hinzuzufügen. `ulimits` Geben Sie die Werte Name, Soft Limit und Hard Limit ein und wählen Sie dann Add ulimit aus.

q. Wählen Sie für Abhängigkeiten — optional die Option Container-Abhängigkeiten hinzufügen aus. Wählen Sie den Namen des Containers und seinen Status aus, um zu bestimmen, wann dieser Container gestartet wird.

17. Wenn Sie nur einen Container konfiguriert haben, müssen Sie Container hinzufügen auswählen und die Konfiguration des neuen Containers abschließen. Andernfalls wählen Sie Weiter zur Überprüfung aus.

Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten

Bevor Sie Jobs in ausführen können AWS Batch, müssen Sie eine Jobdefinition erstellen. Dieser Prozess unterscheidet sich geringfügig zwischen parallel Aufträgen mit einem Knoten und mehreren Knoten. In diesem Thema wird speziell behandelt, wie Sie eine Jobdefinition für einen parallel Job mit AWS Batch mehreren Knoten erstellen (auch bekannt als Gruppenplanung). Weitere Informationen finden Sie unter parallel Jobs mit mehreren Knoten.

> **Note**
>
> AWS Fargate unterstützt keine parallel Jobs mit mehreren Knoten.

Inhalt

- Tutorial: Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten auf Amazon-Ressourcen EC2

Tutorial: Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten auf Amazon-Ressourcen EC2

Um eine parallel Jobdefinition mit mehreren Knoten auf Amazon Elastic Compute Cloud (Amazon EC2) -Ressourcen zu erstellen.

> **Note**
>
> Informationen zum Erstellen einer Auftragsdefinition mit einem Knoten finden Sie unter. Tutorial: Erstellen Sie eine Jobdefinition mit einem Knoten auf Amazon-Ressourcen EC2

So erstellen Sie eine parallel Jobdefinition mit mehreren Knoten auf EC2 Amazon-Ressourcen:

1. Öffnen Sie die AWS Batch Konsole unter. https://console.aws.amazon.com/batch/
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich die Option Jobdefinitionen aus.
4. Wählen Sie Erstellen aus.
5. Wählen Sie als Orchestrierungstyp Amazon Elastic Compute Cloud (Amazon EC2).
6. Aktivieren Sie für Multi-Node-Parallel aktivieren die Option Multi-Node-Parallel.
7. Geben Sie unter Name einen eindeutigen Namen für Ihre Jobdefinition ein. Der Name kann bis zu 128 Zeichen lang sein und Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
8. (Optional) Geben Sie für das Ausführungstimeout die maximale Anzahl von Sekunden an, für die Auftragsversuche ausgeführt werden sollen. Wenn ein Versuch die Timeoutdauer überschreitet, wird der Versuch gestoppt und erhält einen `FAILED` Status. Weitere Informationen finden Sie unter Timeouts bei der Job.
9. (Optional) Aktivieren Sie die Zeitplanungspriorität. Geben Sie einen Wert für die Planungspriorität zwischen 0 und 100 ein. Höheren Werten wird eine höhere Priorität gegenüber niedrigeren Werten eingeräumt.
10. (Optional) Geben Sie für Auftragsversuche ein, wie oft AWS Batch versucht wird, den Job in den `RUNNABLE` Status zu versetzen. Geben Sie eine Zahl zwischen 1 und 10 ein.
11. (Optional) Wählen Sie unter Bedingungen für die Wiederholungsstrategie die Option Beim Beenden bewerten hinzufügen aus. Geben Sie mindestens einen Parameterwert ein und wählen

Sie dann eine Aktion aus. Für jeden Satz von Bedingungen muss Aktion entweder auf „Erneut versuchen“ oder „Beenden“ gesetzt sein. Diese Aktionen bedeuten Folgendes:

- Wiederholen — AWS Batch Wiederholt die Versuche, bis die von Ihnen angegebene Anzahl von Auftragsversuchen erreicht ist.
- Beenden — AWS Batch beendet die Wiederholung des Auftrags.

> **Important**
>
> Wenn Sie „Beim Beenden bewerten hinzufügen“ wählen, müssen Sie mindestens einen Parameter konfigurieren und entweder eine Aktion auswählen oder „Beim Beenden auswerten entfernen“ wählen.

12. (Optional) Erweitern Sie „Tags“ und wählen Sie dann „Tag hinzufügen“, um der Ressource Tags hinzuzufügen. Geben Sie einen Schlüssel und einen optionalen Wert ein und wählen Sie dann Tag hinzufügen aus. Sie können auch Tags propagieren aktivieren, um Tags aus dem Job und der Auftragsdefinition an die Amazon ECS-Aufgabe weiterzugeben.
13. Wählen Sie „Nächste Seite“.
14. Geben Sie unter Number of Nodes (Anzahl von Knoten) die Gesamtanzahl von Knoten ein, die für Ihren Auftrag verwendet werden sollen.
15. Unter Main node (Hauptknoten) geben Sie den für den Hauptknoten zu verwendenden Knotenindex ein. Der standardmäßige Hauptknotenindex lautet `0`.
16. Wählen Sie für Instance-Typ einen Instance-Typ aus.

> **Note**
>
> Der von Ihnen gewählte Instance-Typ gilt für alle Knoten.

17. Wählen Sie für Parameter die Option Parameter hinzufügen aus, um Platzhalter für die Parametersetzung als Schlüssel - und optionale Wertepaare hinzuzufügen.
18. Gehen Sie im Abschnitt Knotenbereiche wie folgt vor:

    a. Wählen Sie Knotenbereich hinzufügen aus. Dadurch wird ein Abschnitt „Knotenbereich“ erstellt.

    b. Legen Sie unter Target nodes (Zielknoten) mit der Notation `range_start:range_end` den Bereich für Ihre Knotengruppe fest.

Sie können bis zu fünf Knotenbereiche für die Knoten erstellen, die Sie für Ihren Job angegeben haben. Knotenbereiche verwenden den Indexwert für einen Knoten und der Knotenindex beginnt bei 0. Stellen Sie sicher, dass der Indexwert für das Bereichsende Ihrer letzten Knotengruppe um eins unter der von Ihnen angegebenen Anzahl von Knoten liegt. Nehmen wir beispielsweise an, Sie haben 10 Knoten angegeben und möchten eine einzelne Knotengruppe verwenden. Dann ist Ihr Endbereich 9.

c. Wählen Sie unter Bild das Docker Bild aus, das Sie für Ihren Job verwenden möchten. Images in der Docker Hub-Registrierung sind standardmäßig verfügbar. Sie können auch andere Repositorys mit `repository-url/image:tag` angeben. Der Name kann bis zu 225 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-), Unterstriche (_), Doppelpunkte (:), Schrägstriche (/) und Zahlenzeichen (#) enthalten. Dieser Parameter wird `Image` zugeordnet im Abschnitt Create a container (Erstellen eines Containers) im Docker-Remote-API und dem `IMAGE`-Parameter von docker run.

> **Note**
>
> Die Docker-Image-Architektur muss der Prozessorarchitektur der Compute-Ressourcen entsprechen, für die sie geplant sind. So können beispielsweise ARM-basierte Docker-Images nur auf ARM-basierten Compute-Ressourcen ausgeführt werden.

- Bilder in öffentlichen Amazon ECR-Repositorys verwenden die vollständigen `registry/repository[:tag]` oder die `registry/repository[@digest]` Namenskonventionen (z. B.`public.ecr.aws/registry_alias/my-web-app:latest`).
- Bilder in Amazon ECR-Repositorys verwenden die vollständige `registry/repository[:tag]` Namenskonvention. Beispiel: `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`
- Images in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B `amazon/amazon-ecs-agent`).
- Image in anderen Online-Repositorys sind durch einen Domain-Namen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

d. Geben Sie für Command die Befehle als entsprechende JSON-String-Array-Entsprechung in das Feld ein.

   Dieser Parameter ist `Cmd` im Abschnitt Erstellen eines Containers der Docker Remote-API und dem Parameter `COMMAND` von docker run zugeordnet. Weitere Informationen zu dem Docker `CMD` Parameter finden Sie unter https://docs.docker.com/engine/reference/builder/#cmd.

   > **Note**
   >
   > Sie können Standardwerte für die Parameterersetzung und Platzhalter in Ihrem Befehl verwenden. Weitere Informationen finden Sie unter Parameter.

e. Geben Sie für v die Anzahl von v anCPUs, die für CPUs den Container reserviert werden sollen. Dieser Parameter ordnet zu `CpuShares` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--cpu-shares` für die docker run zu. Jede vCPU entspricht 1.024 CPU-Anteilen. Sie müssen mindestens eine vCPU angeben.

f. Geben Sie unter Arbeitsspeicher die Arbeitsspeichergrenze (in MiB) für die Präsentation an den Container des Auftrags an. Wenn Ihr Container versucht, den hier angegebenen Speicher zu überschreiten, wird der Container gestoppt. Dieser Parameter ordnet zu `Memory` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--memory` für die docker run zu. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen.

   > **Note**
   >
   > Um Ihre Ressourcennutzung zu maximieren, können Sie Ihren Jobs so viel Speicher wie möglich für einen bestimmten Instance-Typ zur Verfügung stellen. Weitere Informationen finden Sie unter Speicherverwaltung für Rechenressourcen.

g. (Optional) Geben Sie unter Anzahl von GPUs die Anzahl GPUs Ihrer Job-Nutzungen an. Der Job wird auf einem Container mit der angegebenen Anzahl von Containern ausgeführt GPUs , die an diesen Container angeheftet sind.

h. (Optional) Für die Rolle Job können Sie eine IAM-Rolle angeben, die dem Container in Ihrem Job Berechtigungen zur Verwendung von erteilt. AWS APIs Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen, einschließlich der

Voraussetzungen für die Konfiguration, finden Sie unter IAM Roles for Tasks im Amazon Elastic Container Service Developer Guide.

> **Note**
>
> Für Jobs, die auf Fargate-Ressourcen ausgeführt werden, ist eine Jobrolle erforderlich.

> **Note**
>
> Hier werden nur Rollen angezeigt, für die die Amazon Elastic Container Service Task Role Vertrauensstellung besteht. Weitere Informationen zum Erstellen einer IAM-Rolle für Ihre AWS Batch Jobs finden Sie unter Creating an IAM Role and Policy for your Tasks im Amazon Elastic Container Service Developer Guide.

i. (Optional) Geben Sie für die Ausführungsrolle eine IAM-Rolle an, die den Amazon ECS-Container-Agenten die Erlaubnis erteilt, AWS API-Aufrufe in Ihrem Namen durchzuführen. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen finden Sie unter IAM-Rollen für die Ausführung von Amazon ECS-Aufgaben im Amazon Elastic Container Service Developer Guide.

19. (Optional) Erweitern Sie Zusätzliche Konfiguration:

    a. Wählen Sie für Umgebungsvariablen die Option Umgebungsvariable hinzufügen aus, um Umgebungsvariablen als Name-Wert-Paare hinzuzufügen. Diese Variablen werden an den Container übergeben.

    b. Für die Job-Rollenkonfiguration können Sie eine IAM-Rolle angeben, die dem Container in Ihrem Job Berechtigungen zur Verwendung von gewährt. AWS APIs Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen, einschließlich der Voraussetzungen für die Konfiguration, finden Sie unter IAM Roles for Tasks im Amazon Elastic Container Service Developer Guide.

    > **Note**
    >
    > Für Jobs, die auf Fargate-Ressourcen ausgeführt werden, ist eine Jobrolle erforderlich.

> **Note**
>
> Hier werden nur Rollen angezeigt, für die die Amazon Elastic Container Service Task Role Vertrauensstellung besteht. Weitere Informationen zum Erstellen einer IAM-Rolle für Ihre AWS Batch Jobs finden Sie unter Creating an IAM Role and Policy for your Tasks im Amazon Elastic Container Service Developer Guide.

c. Geben Sie für die Ausführungsrolle eine IAM-Rolle an, die den Amazon ECS-Container-Agenten die Erlaubnis erteilt, AWS API-Aufrufe in Ihrem Namen durchzuführen. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen finden Sie unter IAM-Rollen für die Ausführung von Amazon ECS-Aufgaben im Amazon Elastic Container Service Developer Guide.

20. Im Abschnitt Sicherheitskonfiguration:

    a. (Optional) Um dem Container Ihres Jobs erhöhte Rechte auf der Host-Instance zu gewähren (ähnlich wie dem `root` Benutzer), aktivieren Sie Privilegiert. Dieser Parameter ordnet zu `Privileged` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--privileged` für die docker run zu.

    b. (Optional) Geben Sie unter Benutzer den Benutzernamen ein, der im Container verwendet werden soll. Dieser Parameter ordnet zu `User` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--user` für die docker run zu.

    c. (Optional) Wählen Sie unter Secrets die Option Add secret aus, um Secrets als Name-Wert-Paare hinzuzufügen. Diese Geheimnisse werden im Container offengelegt. Weitere Informationen finden Sie unter:SecretOptionsLogConfiguration.

21. Im Abschnitt zur Linux-Konfiguration:

    a. Aktivieren Sie die Option Schreibgeschütztes Dateisystem aktivieren, um den Schreibzugriff auf das Volume zu unterbinden.

    b. (Optional) Aktivieren Sie die Option „initProzess aktivieren“, um einen `init` Prozess innerhalb des Containers auszuführen. Dieser Prozess leitet Signale weiter und verarbeitet Prozesse.

    c. Geben Sie für Größe des gemeinsamen Speichers die Größe (in MiB) des `/dev/shm` Volumes ein.

d. Geben Sie für Max. Swap-Größe die Gesamtmenge an Swap-Speicher (in MiB) ein, die der Container verwenden kann.

e. Geben Sie für Swappiness einen Wert zwischen 0 und 100 ein, um das Swappiness-Verhalten des Containers anzugeben. Wenn Sie keinen Wert angeben und Swapping aktiviert ist, wird der Standardwert auf 60 gesetzt. Weitere Informationen finden Sie unter:Swappiness. LinuxParameters

f. (Optional) Wählen Sie für Geräte die Option Gerät hinzufügen aus, um ein Gerät hinzuzufügen:

   i. Geben Sie für Container path (Container-Pfad) den Pfad in der Container-Instance an, um das der Host-Instance zugeordnete Gerät zugänglich zu machen. Wenn Sie dieses Feld leer lassen, wird der Hostpfad im Container verwendet.

   ii. Geben Sie für Host path (Host-Pfad) den Pfad eines Geräts in der Host-Instance an.

   iii. Wählen Sie unter Berechtigungen eine oder mehrere Berechtigungen aus, die auf das Gerät angewendet werden sollen. Die verfügbaren Berechtigungen sind READ, WRITE und MKNOD.

22. (Optional) Wählen Sie für Bereitstellungspunkte die Option Konfiguration für Bereitstellungspunkte hinzufügen aus, um Bereitstellungspunkte für Datenvolumes hinzuzufügen. Sie müssen das Quellvolume und den Containerpfad angeben. Diese Einhängepunkte werden an den Docker Daemon auf einer Container-Instance übergeben. Sie können sich auch dafür entscheiden, das Volume schreibgeschützt zu machen.

23. (Optional) Wählen Sie für die Ulimits-Konfiguration die Option Ulimit hinzufügen aus, um einen `ulimits` Wert für den Container hinzuzufügen. Geben Sie die Werte Name, Soft Limit und Hard Limit ein und wählen Sie dann Add ulimit aus.

24. (Optional) Wählen Sie für die Volumes-Konfiguration die Option Volume hinzufügen aus, um eine Liste von Volumes zu erstellen, die an den Container übergeben werden sollen. Geben Sie den Namen und den Quellpfad für das Volume ein und wählen Sie dann Volume hinzufügen. Sie können sich auch dafür entscheiden, EFS aktivieren zu aktivieren.

25. (Optional) Wählen Sie für Tmpfs Add tmpfs aus, um einen Mount hinzuzufügen. `tmpfs`

26. Gehen Sie im Abschnitt „Aufgabeneigenschaften“ wie folgt vor:

    a. Wählen Sie für Ausführungsrolle — bedingt eine Rolle aus, damit Amazon ECS-Agenten AWS API-Aufrufe in Ihrem Namen tätigen können. Weitere Informationen zur Erstellung einer Ausführungsrolle finden Sie unterTutorial: Erstellen Sie die IAM-Ausführungsrolle.

b.

> **Important**
>
> Um den ECS-Ausführungsbefehl verwenden zu können, muss Ihre Rechenumgebung die Überlegungen zur Rechenumgebung für parallel Jobs mit mehreren Knoten erfüllen.

Wählen Sie den ECS-Ausführungsbefehl aktivieren, um den direkten Zugriff auf die Amazon ECS-Container-Shell zu aktivieren und das Host-Betriebssystem zu umgehen. Sie müssen eine Aufgabenrolle wählen.

> **Important**
>
> Der ECS-Befehl Execute setzt voraus, dass das Dateisystem beschreibbar ist.

c. Wählen Sie für Aufgabenrolle eine Amazon ECS Identity and Access Management (IAM) -Rolle aus, damit der Container AWS API-Aufrufe in Ihrem Namen tätigen kann. Weitere Informationen finden Sie unter Amazon ECS-Aufgaben-IAM-Rolle im Amazon Elastic Container Service Developer Guide.

27. (Optional) Gehen Sie im Abschnitt zur Konfiguration der Protokollierung wie folgt vor:

a. Wählen Sie unter Protokolltreiber den Protokolltreiber aus, den Sie verwenden möchten. Weitere Informationen zu den verfügbaren Protokolltreibern finden Sie unter:LogDriverLogConfiguration.

> **Note**
>
> Standardmäßig wird der `awslogs` Protokolltreiber verwendet.

b. Wählen Sie unter Optionen die Option Hinzufügen aus, um eine Option hinzuzufügen. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Option hinzufügen.

c. Wählen Sie für Secrets die Option Add Secret aus. Geben Sie ein Name-Wert-Paar ein und wählen Sie dann Geheimnis hinzufügen aus, um ein Geheimnis hinzuzufügen.

> **Tip**
>
> Weitere Informationen finden Sie unter:Geheime Optionen. LogConfiguration

28. Wählen Sie „Nächste Seite“.
29. Lesen Sie zur Überprüfung der Jobdefinition die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Jobdefinition erstellen.

Referenz: Jobdefinitionsvorlage, die verwendet ContainerProperties

Im Folgenden finden Sie eine leere Jobdefinitionsvorlage, die einen einzelnen Container enthält. Sie können diese Vorlage verwenden, um Ihre Jobdefinition zu erstellen, die dann in einer Datei gespeichert und mit der AWS CLI `--cli-input-json` Option verwendet werden kann. Weitere Informationen zu diesen Parametern finden Sie unter JobDefinition.

> **Note**
>
> Mit dem folgenden AWS CLI Befehl können Sie eine Jobdefinitionsvorlage für einen einzelnen Container generieren:
>
> ```
> $ aws batch register-job-definition --generate-cli-skeleton
> ```

```
{
    "jobDefinitionName": "",
    "type": "container",
    "parameters": {
        "KeyName": ""
    },
    "schedulingPriority": 0,
    "containerProperties": {
        "image": "",
        "vcpus": 0,
        "memory": 0,
        "command": [
            ""
        ],
        "jobRoleArn": "",
        "executionRoleArn": "",
        "volumes": [
            {
                "host": {
```

```
                    "sourcePath": ""
                },
                "name": "",
                "efsVolumeConfiguration": {
                    "fileSystemId": "",
                    "rootDirectory": "",
                    "transitEncryption": "ENABLED",
                    "transitEncryptionPort": 0,
                    "authorizationConfig": {
                        "accessPointId": "",
                        "iam": "DISABLED"
                    }
                }
            }
        ],
        "environment": [
            {
                "name": "",
                "value": ""
            }
        ],
        "mountPoints": [
            {
                "containerPath": "",
                "readOnly": true,
                "sourceVolume": ""
            }
        ],
        "readonlyRootFilesystem": true,
        "privileged": true,
        "ulimits": [
            {
                "hardLimit": 0,
                "name": "",
                "softLimit": 0
            }
        ],
        "user": "",
        "instanceType": "",
        "resourceRequirements": [
            {
                "value": "",
                "type": "MEMORY"
            }
```

```
        ],
        "linuxParameters": {
            "devices": [
                {
                    "hostPath": "",
                    "containerPath": "",
                    "permissions": [
                        "WRITE"
                    ]
                }
            ],
            "initProcessEnabled": true,
            "sharedMemorySize": 0,
            "tmpfs": [
                {
                    "containerPath": "",
                    "size": 0,
                    "mountOptions": [
                        ""
                    ]
                }
            ],
            "maxSwap": 0,
            "swappiness": 0
        },
        "logConfiguration": {
            "logDriver": "syslog",
            "options": {
                "KeyName": ""
            },
            "secretOptions": [
                {
                    "name": "",
                    "valueFrom": ""
                }
            ]
        },
        "secrets": [
            {
                "name": "",
                "valueFrom": ""
            }
        ],
        "networkConfiguration": {
```

```
            "assignPublicIp": "DISABLED"
        },
        "fargatePlatformConfiguration": {
            "platformVersion": ""
        }
    },
    "nodeProperties": {
        "numNodes": 0,
        "mainNode": 0,
        "nodeRangeProperties": [
            {
                "targetNodes": "",
                "container": {
                    "image": "",
                    "vcpus": 0,
                    "memory": 0,
                    "command": [
                        ""
                    ],
                    "jobRoleArn": "",
                    "executionRoleArn": "",
                    "volumes": [
                        {
                            "host": {
                                "sourcePath": ""
                            },
                            "name": "",
                            "efsVolumeConfiguration": {
                                "fileSystemId": "",
                                "rootDirectory": "",
                                "transitEncryption": "DISABLED",
                                "transitEncryptionPort": 0,
                                "authorizationConfig": {
                                    "accessPointId": "",
                                    "iam": "ENABLED"
                                }
                            }
                        }
                    ],
                    "environment": [
                        {
                            "name": "",
                            "value": ""
                        }
```

```
                    ],
                    "mountPoints": [
                        {
                            "containerPath": "",
                            "readOnly": true,
                            "sourceVolume": ""
                        }
                    ],
                    "readonlyRootFilesystem": true,
                    "privileged": true,
                    "ulimits": [
                        {
                            "hardLimit": 0,
                            "name": "",
                            "softLimit": 0
                        }
                    ],
                    "user": "",
                    "instanceType": "",
                    "resourceRequirements": [
                        {
                            "value": "",
                            "type": "MEMORY"
                        }
                    ],
                    "linuxParameters": {
                        "devices": [
                            {
                                "hostPath": "",
                                "containerPath": "",
                                "permissions": [
                                    "WRITE"
                                ]
                            }
                        ],
                        "initProcessEnabled": true,
                        "sharedMemorySize": 0,
                        "tmpfs": [
                            {
                                "containerPath": "",
                                "size": 0,
                                "mountOptions": [
                                    ""
                                ]
```

```
                        }
                    ],
                    "maxSwap": 0,
                    "swappiness": 0
                },
                "logConfiguration": {
                    "logDriver": "awslogs",
                    "options": {
                        "KeyName": ""
                    },
                    "secretOptions": [
                        {
                            "name": "",
                            "valueFrom": ""
                        }
                    ]
                },
                "secrets": [
                    {
                        "name": "",
                        "valueFrom": ""
                    }
                ],
                "networkConfiguration": {
                    "assignPublicIp": "DISABLED"
                },
                "fargatePlatformConfiguration": {
                    "platformVersion": ""
                }
            }
        }
    ]
},
"retryStrategy": {
    "attempts": 0,
    "evaluateOnExit": [
        {
            "onStatusReason": "",
            "onReason": "",
            "onExitCode": "",
            "action": "RETRY"
        }
    ]
},
```

```
    "propagateTags": true,
    "timeout": {
        "attemptDurationSeconds": 0
    },
    "tags": {
        "KeyName": ""
    },
    "platformCapabilities": [
        "EC2"
    ],
    "eksProperties": {
        "podProperties": {
            "serviceAccountName": "",
            "hostNetwork": true,
            "dnsPolicy": "",
            "containers": [
                {
                    "name": "",
                    "image": "",
                    "imagePullPolicy": "",
                    "command": [
                        ""
                    ],
                    "args": [
                        ""
                    ],
                    "env": [
                        {
                            "name": "",
                            "value": ""
                        }
                    ],
                    "resources": {
                        "limits": {
                            "KeyName": ""
                        },
                        "requests": {
                            "KeyName": ""
                        }
                    },
                    "volumeMounts": [
                        {
                            "name": "",
                            "mountPath": "",
```

```
                            "readOnly": true
                        }
                    ],
                    "securityContext": {
                        "runAsUser": 0,
                        "runAsGroup": 0,
                        "privileged": true,
                        "readOnlyRootFilesystem": true,
                        "runAsNonRoot": true
                    }
                }
            ],
            "volumes": [
                {
                    "name": "",
                    "hostPath": {
                        "path": ""
                    },
                    "emptyDir": {
                        "medium": "",
                        "sizeLimit": ""
                    },
                    "secret": {
                        "secretName": "",
                        "optional": true
                    }
                }
            ]
        }
    }
}
```

Referenz: Jobdefinitionsparameter für ContainerProperties

Jobdefinitionen, die ich verwende, `ContainerProperties`sind in mehrere Teile aufgeteilt:

- der Name der Jobdefinition
- der Typ der Jobdefinition
- Die Standardwerte für den Platzhalter für die Parameterersetzung
- die Container-Eigenschaften für den Job

- die Amazon EKS-Eigenschaften für die Auftragsdefinition, die für Jobs erforderlich sind, die auf Amazon EKS-Ressourcen ausgeführt werden
- die Knoteneigenschaften, die für einen Paralleljob mit mehreren Knoten erforderlich sind
- die Plattformfunktionen, die für Jobs erforderlich sind, die auf Fargate-Ressourcen ausgeführt werden
- die standardmäßigen Tag-Propagierungsdetails der Jobdefinition
- die standardmäßige Wiederholungsstrategie für die Auftragsdefinition
- die standardmäßige Planungspriorität für die Auftragsdefinition
- die Standard-Tags für die Jobdefinition
- das Standard-Timeout für die Jobdefinition

Inhalt

- Name der Jobdefinition
- Typ
- Parameter
- Eigenschaften des Containers
- Amazon EKS-Eigenschaften
- Funktionen der Plattform
- Tags weitergeben
- Eigenschaften des Knotens
- Strategie erneut versuchen
- Priorität bei der Planung
- Tags
- Timeout

Name der Jobdefinition

`jobDefinitionName`

Wenn Sie eine Auftragsdefinition registrieren, geben Sie einen Namen ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten. Die erste Jobdefinition, die mit diesem Namen registriert ist, erhält die

Revision 1. Alle nachfolgenden Auftragsdefinitionen, die mit diesem Namen registriert werden, erhalten eine inkrementelle Versionsnummer.

Typ: Zeichenfolge

Erforderlich: Ja

Typ

`type`

Wenn Sie eine Auftragsdefinition registrieren, geben Sie den Auftragstyp an. Wenn der Job auf Fargate-Ressourcen ausgeführt wird, wird er `multinode` nicht unterstützt. Weitere Informationen zu parallelen Aufträgen mit mehreren Knoten finden Sie unter Erstellen Sie eine parallel Jobdefinition mit mehreren Knoten.

Typ: Zeichenfolge

Zulässige Werte: `container` | `multinode`

Erforderlich: Ja

Parameter

`parameters`

Wenn Sie einen Job einreichen, können Sie Parameter angeben, die die Platzhalter ersetzen oder die Standardparameter der Jobdefinition überschreiben. Die Parameter in Anforderungen zum Senden von Aufträgen haben Vorrang vor den Standardeinstellungen in einer Auftragsdefinition. Das bedeutet, dass Sie dieselbe Jobdefinition für mehrere Jobs verwenden können, die dasselbe Format verwenden. Sie können Werte im Befehl auch während der Übermittlung programmgesteuert ändern.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Erforderlich: Nein

Wenn Sie eine Auftragsdefinition registrieren, können Sie Parameterersetzungsplatzhalter in das Feld `command` der Container-Eigenschaften eines Auftrags verwenden. Die Syntax ist wie folgt.

```
"command": [
    "ffmpeg",
    "-i",
    "Ref::inputfile",
    "-c",
    "Ref::codec",
    "-o",
    "Ref::outputfile"
]
```

Im obigen Beispiel befinden sich die Parameterersetzungsplatzhalter `Ref::inputfile`, `Ref::codec` und `Ref::outputfile` im Befehl. Sie können das `parameters` Objekt in der Jobdefinition verwenden, um Standardwerte für diese Platzhalter festzulegen. Um beispielsweise einen Standard für die `Ref::codec`-Platzhalter einzustellen, legen Sie Folgendes in der Auftragsdefinition fest:

```
"parameters" : {"codec" : "mp4"}
```

Wenn diese Auftragsdefinition zur Ausführung übergeben wird, wird das `Ref::codec` Argument im Befehl für den Container durch den Standardwert ersetzt. `mp4`

Eigenschaften des Containers

Wenn Sie eine Jobdefinition registrieren, geben Sie eine Liste von Container-Eigenschaften an, die bei der Auftragserteilung an den Docker-Daemon auf einer Container-Instance übergeben werden. Die folgenden Container-Eigenschaften sind in einer Auftragsdefinition zulässig. Für Aufträge mit einzelnen Knoten werden diese Container-Eigenschaften auf Auftragsdefinitionsebene festgelegt. Für parallele Aufträge mit mehreren Knoten werden die Container-Eigenschaften für jede Knotengruppe auf der Ebene der Eigenschaften des Knotens festgelegt.

`command`

Der Befehl, der an den Container übergeben wird. Dieser Parameter ist `Cmd` im Abschnitt Erstellen eines Containers der Docker Remote-API und dem Parameter `COMMAND` von docker run zugeordnet. Weitere Informationen zum `CMD` Docker-Parameter finden Sie unter https://docs.docker.com/engine/ reference/builder/ #cmd.

```
"command": ["string", ...]
```

Typ: Zeichenfolgen-Array

Erforderlich: Nein

`environment`

Die Umgebungsvariablen, die an einen Container übergeben werden. Dieser Parameter ordnet zu `Env` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--env` für die docker run zu.

> **Important**
>
> Wir raten davon ab, Umgebungsvariablen im Klartext für sensible Informationen wie Anmeldeinformationen zu verwenden.

> **Note**
>
> Umgebungsvariablen dürfen nicht mit beginnen. `AWS_BATCH` Diese Namenskonvention ist Variablen vorbehalten, die vom AWS Batch Dienst festgelegt werden.

Typ: Array von Schlüssel-Wert-Paaren

Erforderlich: Nein

`name`

Der Name der Umgebungsvariable.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `environment` verwendet wird.

`value`

Der Wert der Umgebungsvariable.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `environment` verwendet wird.

```
"environment" : [
```

```
    { "name" : "envName1", "value" : "envValue1" },
    { "name" : "envName2", "value" : "envValue2" }
]
```

`executionRoleArn`

Wenn Sie eine Jobdefinition registrieren, können Sie eine IAM-Rolle angeben. Die Rolle gewährt dem Amazon ECS-Container-Agenten die Erlaubnis, die API-Aktionen, die in den zugehörigen Richtlinien angegeben sind, in Ihrem Namen aufzurufen. Jobs, die auf Fargate-Ressourcen ausgeführt werden, müssen eine Ausführungsrolle bereitstellen. Weitere Informationen finden Sie unter AWS Batch IAM-Ausführungsrolle.

Typ: Zeichenfolge

Erforderlich: Nein

`fargatePlatformConfiguration`

Die Plattformkonfiguration für Jobs, die auf Fargate-Ressourcen ausgeführt werden. Jobs, die auf EC2 Ressourcen ausgeführt werden, dürfen diesen Parameter nicht angeben.

Typ: FargatePlatformConfiguration Objekt

Erforderlich: Nein

`platformVersion`

Die AWS Fargate-Plattformversion, die für die Jobs verwendet wird, oder `LATEST` um eine aktuelle, zugelassene Version der AWS Fargate-Plattform zu verwenden.

Typ: Zeichenfolge

Standard: `LATEST`

Erforderlich: Nein

`image`

Das Bild, mit dem ein Job gestartet wurde. Diese Zeichenfolge wird direkt an den Docker-Daemon übergeben. Images in der Docker Hub-Registrierung sind standardmäßig verfügbar. Sie können auch andere Repositorys mit *`repository-url`*`/`*`image`*`:`*`tag`* angeben. Bis zu 255 Buchstaben (Groß- und Kleinbuchstaben), Ziffern, Bindestriche, Unterstriche, Doppelpunkte, Punkte und

Schrägstriche sind zulässig. Dieser Parameter wird `Image` zugeordnet im Abschnitt Create a container (Erstellen eines Containers) im Docker-Remote-API und dem `IMAGE`-Parameter von docker run.

> **Note**
>
> Die Docker-Image-Architektur muss der Prozessorarchitektur der Compute-Ressourcen entsprechen, für die sie geplant sind. So können beispielsweise ARM-basierte Docker-Images nur auf ARM-basierten Compute-Ressourcen ausgeführt werden.

- Bilder in öffentlichen Amazon ECR-Repositorys verwenden die vollständigen `registry/repository[:tag]` oder die `registry/repository[@digest]` Namenskonventionen (z. B.`public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*).
- Bilder in Amazon ECR-Repositorys verwenden die vollständige `registry/repository:[tag]` Namenskonvention. Beispiel, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*.
- Images in offiziellen Repositorys in Docker Hub verwenden einen einzelnen Namen (z. B. `ubuntu` oder `mongo`).
- Images in anderen Repositorys in Docker Hub sind mit einem Organisationsnamen qualifiziert (z. B, `amazon/amazon-ecs-agent`).
- Image in anderen Online-Repositorys sind durch einen Domain-Namen zusätzlich qualifiziert (z. B. `quay.io/assemblyline/ubuntu`).

Typ: Zeichenfolge

Erforderlich: Ja

`instanceType`

Der Instance-Typ zur Verwendung für einen parallelen Auftrag mit mehreren Knoten. Alle Knotengruppen müssen in einem parallelen Auftrag mit mehreren Knoten den gleichen Instance-Typ verwenden. Dieser Parameter ist nicht gültig für Container-Jobs mit einem Knoten oder für Jobs, die auf Fargate-Ressourcen ausgeführt werden.

Typ: Zeichenfolge

Erforderlich: Nein

`jobRoleArn`

Wenn Sie eine Jobdefinition registrieren, können Sie eine IAM-Rolle angeben. Die Rolle stellt dem Auftrags-Container Berechtigungen zum Aufrufen der API-Aktionen zur Verfügung, die in den zugehörigen Richtlinien in Ihrem Namen festgelegt sind. Weitere Informationen finden Sie unter IAM-Rollen für Aufgaben im Entwicklerhandbuch zum Amazon Elastic Container Service.

Typ: Zeichenfolge

Erforderlich: Nein

`linuxParameters`

Linux-spezifische Änderungen, die auf den Container angewendet werden, z. B. Details für Gerätezuordnungen.

```
"linuxParameters": {
    "devices": [
        {
            "hostPath": "string",
            "containerPath": "string",
            "permissions": [
                "READ", "WRITE", "MKNOD"
            ]
        }
    ],
    "initProcessEnabled": true|false,
    "sharedMemorySize": 0,
    "tmpfs": [
        {
            "containerPath": "string",
            "size": integer,
            "mountOptions": [
                "string"
            ]
        }
    ],
    "maxSwap": integer,
    "swappiness": integer
}
```

Typ: LinuxParameters Objekt

Erforderlich: Nein

`devices`

Liste der im Container zugeordneten Geräte. Dieser Parameter ist `Devices` im Abschnitt Einen Container erstellen der Docker-Remote-API und der Option `--device` für die Docker-Ausführung zugeordnet.

> **Note**
>
> Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden.

Typ: Array von Device-Objekten

Erforderlich: Nein

`hostPath`

Pfad, in dem sich das in der Host-Container-Instance verfügbare Gerät befindet.

Typ: Zeichenfolge

Erforderlich: Ja

`containerPath`

Pfad, in dem das Gerät im Container verfügbar gemacht wird. Wenn dies nicht angegeben ist, wird das Gerät unter demselben Pfad wie der Hostpfad verfügbar gemacht.

Typ: Zeichenfolge

Erforderlich: Nein

`permissions`

Berechtigungen für das Gerät im Container. Wenn dies nicht angegeben ist, sind die Berechtigungen auf `READWRITE`, und gesetzt`MKNOD`.

Typ: Zeichenfolgen-Array

Erforderlich: Nein

Zulässige Werte: `READ` | `WRITE` | `MKNOD`

`initProcessEnabled`

Wenn der Wert "true" lautet, führen Sie einen `init`-Prozess innerhalb des Containers aus, der Signalen weiterleitet und Prozesse aufnimmt. Dieser Parameter wird der Option `--init` für die Docker-Ausführung zugeordnet. Für diesen Parameter muss Ihre Docker Remote API Version 1.25 oder höher in Ihrer Container-Instance verwenden. Um die Docker Remote API-Version auf Ihrer Container-Instance zu überprüfen, melden Sie sich bei Ihrer Container-Instance an und führen Sie den folgenden Befehl aus: `sudo docker version | grep "Server API version"`

Typ: Boolesch

Erforderlich: Nein

`maxSwap`

Die Gesamtmenge an Swap-Speicher (in MiB), die ein Job verwenden kann. Dieser Parameter ist in die Option `--memory-swap` zur Docker-Ausführung übersetzt, wobei der Wert die Summe aus dem Container-Speicher und dem Wert `maxSwap` ist. Weitere Informationen finden Sie unter `--memory-swap`-Details in der Docker-Dokumentation.

Wenn als `maxSwap`-Wert `0` angegeben wird, verwendet der Container keine Auslagerung. Zulässige Werte sind `0` oder eine beliebige positive Ganzzahl. Wenn der `maxSwap` Parameter weggelassen wird, verwendet der Container die Swap-Konfiguration für die Container-Instance, auf der er ausgeführt wird. Es muss ein Wert für `maxSwap` festgelegt werden, damit der Parameter `swappiness` verwendet werden kann.

> **Note**
>
> Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden.

Typ: Ganzzahl

Erforderlich: Nein

`sharedMemorySize`

Der Wert für die Größe des `/dev/shm`-Volumes (in MiB). Dieser Parameter wird der Option `--shm-size` für die Docker-Ausführung zugeordnet.

> **Note**
>
> Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden.

Typ: Ganzzahl

Erforderlich: Nein

`swappiness`

Auf diese Weise können Sie das Speicherauslagerungsverhalten eines Containers optimieren. Ein `swappiness` Wert von `0` bewirkt, dass ein Austausch nicht stattfindet, sofern dies nicht unbedingt erforderlich ist. Ein `swappiness`-Wert von `100` führt dazu, dass Seiten aggressiv ausgelagert werden. Akzeptierte Werte sind Ganzzahlen zwischen `0` und `100`. Wenn der Parameter `swappiness` nicht angegeben wird, wird der Standardwert `60` verwendet. Wenn kein Wert für `maxSwap` angegeben ist, wird dieser Parameter ignoriert. Wenn `maxSwap` auf 0 gesetzt ist, verwendet der Container kein Auslagern. Dieser Parameter wird der Option `--memory-swappiness` für die Docker-Ausführung zugeordnet.

Beachten Sie Folgendes, wenn Sie eine Auslagerungskonfiguration pro Container verwenden.

- Der Auslagerungsbereich muss auf der Container-Instance aktiviert und zugewiesen werden, damit die Container verwenden können.

  > **Note**
  >
  > Bei Amazon ECS-Optimierungen ist Swap standardmäßig AMIs nicht aktiviert. Sie müssen die Auslagerung auf der Instance aktivieren, um dieses Feature verwenden zu können. Weitere Informationen finden Sie unter Instance Store Swap Volumes im EC2 Amazon-Benutzerhandbuch oder Wie weise ich mithilfe einer Swap-Datei Speicher zu, der als Swap-Speicherplatz in einer EC2 Amazon-Instance verwendet werden soll? .

- Die Swap-Space-Parameter werden nur für Jobdefinitionen unterstützt, die EC2 Ressourcen verwenden.
- Wenn die Container-Definitionsparameter `maxSwap` und `swappiness` in einer Aufgabendefinition weggelassen werden, hat jeder Container den `swappiness`-

Standardwert 60. Die gesamte Swap-Nutzung ist auf das Zweifache der Speicherreservierung des Containers begrenzt.

> **Note**
>
> Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden.

Typ: Ganzzahl

Erforderlich: Nein

`tmpfs`

Der Container-Pfad, Mount-Optionen und Größe des tmpfs-Mounts.

Typ: Array von Tmpfs-Objekten

> **Note**
>
> Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden.

Erforderlich: Nein

`containerPath`

Der absolute Dateipfad in dem Container, in dem das tmpfs-Volume gemountet ist.

Typ: Zeichenfolge

Erforderlich: Ja

`mountOptions`

Die Liste der Mount-Optionen für das tmpfs-Volume.

Gültige Werte: `defaults` "`ro`" | `rw` "|`suid`" | "|`nosuid`" | `dev` "|`nodev`" | `exec` "|`noexec`" | `sync` "|`async`" | "|`dirsync`" | `remount` "|`mand`" | `nomand` "|`atime`" | `noatime` "|`diratime`" | "|`nodiratime`" | `bind` "|`rbind`" | `unbindable` "|`runbindable`" | `private` "|" | `rprivate` "|`shared`" | `rshared` "|`slave`" | `rslave` "|`relatime`" |

`norelatime` "|" `strictatime` "|`nostrictatime`" |“ `mode`"|" `uid` "|" `gid` "|" `nr_inodes` "|" | `nr_blocks` "|"`mpol`“

Typ: Zeichenfolgen-Array

Erforderlich: Nein

`size`

Die Größe des tmpfs-Volumes (in MB).

Typ: Ganzzahl

Erforderlich: Ja

`logConfiguration`

Die Protokollkonfigurationsspezifikation für den Job.

Dieser Parameter ist `LogConfig` im Abschnitt Einen Container erstellen der Docker-Remote-API und der Option `--log-driver` für die Docker-Ausführung zugeordnet. Standardmäßig verwenden Container den gleichen Protokolltreiber wie der Docker-Daemon. Der Container kann jedoch einen anderen Logging-Treiber als der Docker-Daemon verwenden, indem ein Protokolltreiber mit diesem Parameter in der Container-Definition angegeben wird. Um einen anderen Logging-Treiber für einen Container zu verwenden, muss das Protokollsystem entweder auf der Container-Instance oder auf einem anderen Log-Server konfiguriert werden, um Remote-Logging-Optionen bereitzustellen. Weitere Informationen zu den Optionen für die verschiedenen unterstützten Protokolltreiber finden Sie unter Configure logging drivers (Protokolltreiber konfigurieren) in der Docker-Dokumentation.

> **Note**
>
> AWS Batch unterstützt derzeit eine Teilmenge der Protokollierungstreiber, die für den Docker-Daemon verfügbar sind (siehe LogConfigurationDatentyp).

Für diesen Parameter muss Ihre Docker Remote API Version 1.18 oder höher in Ihrer Container-Instance verwenden. Um die Docker Remote API-Version auf Ihrer Container-Instance zu überprüfen, melden Sie sich bei Ihrer Container-Instance an und führen Sie den folgenden Befehl aus: `sudo docker version | grep "Server API version"`

```
"logConfiguration": {
```

```
    "devices": [
        {
            "logDriver": "string",
            "options": {
                "optionName1" : "optionValue1",
                "optionName2" : "optionValue2"
            }
            "secretOptions": [
              {
                  "name" : "secretOptionName1",
                  "valueFrom" : "secretOptionArn1"
              },
              {
                  "name" : "secretOptionName2",
                  "valueFrom" : "secretOptionArn2"
              }
            ]
        }
    ]
}
```

Typ: LogConfiguration Objekt

Erforderlich: Nein

`logDriver`

Der Protokolltreiber, der für den Job verwendet werden soll. AWS Batch Aktiviert standardmäßig den `awslogs` Protokolltreiber. Die für diesen Parameter aufgelisteten gültigen Werte sind Protokolltreiber, mit denen der Amazon-ECS-Container-Agent standardmäßig kommunizieren kann.

Dieser Parameter ist `LogConfig` im Abschnitt Einen Container erstellen der Docker-Remote-API und der Option `--log-driver` für die Docker-Ausführung zugeordnet. Standardmäßig verwenden Jobs denselben Protokollierungstreiber, den der Docker-Daemon verwendet. Der Job kann jedoch einen anderen Logging-Treiber als der Docker-Daemon verwenden, indem ein Protokolltreiber mit diesem Parameter in der Jobdefinition angegeben wird. Wenn Sie einen anderen Protokollierungstreiber für einen Job angeben möchten, muss das Protokollsystem auf der Container-Instance in der Rechenumgebung konfiguriert werden. Oder konfigurieren Sie es alternativ auf einem anderen Protokollserver, um Optionen für die Remote-Protokollierung bereitzustellen. Weitere Informationen zu den Optionen für

die verschiedenen unterstützten Protokolltreiber finden Sie unter Configure logging drivers (Protokolltreiber konfigurieren) in der Docker-Dokumentation.

> **Note**
>
> AWS Batch unterstützt derzeit eine Teilmenge der Protokollierungstreiber, die für den Docker-Daemon verfügbar sind. Es ist möglich, dass in zukünftigen Releases des Amazon-ECS-Container-Agenten weitere Protokolltreiber zur Verfügung stehen.

Die unterstützten Protokolltreiber sind `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog` und `splunk`.

> **Note**
>
> Jobs, die auf Fargate-Ressourcen ausgeführt werden, sind auf die Treiber `awslogs` und die `splunk` Protokolltreiber beschränkt.

Für diesen Parameter muss Ihre Docker Remote API Version 1.18 oder höher in Ihrer Container-Instance verwenden. Um die Docker Remote API-Version auf Ihrer Container-Instance zu überprüfen, melden Sie sich bei Ihrer Container-Instance an und führen Sie den folgenden Befehl aus: `sudo docker version | grep "Server API version"`

> **Note**
>
> Der Amazon ECS-Container-Agent, der auf einer Container-Instance ausgeführt wird, muss die auf dieser Instance verfügbaren Protokollierungstreiber mit der `ECS_AVAILABLE_LOGGING_DRIVERS` Umgebungsvariablen registrieren. Andernfalls können die auf dieser Instance platzierten Container diese Protokollkonfigurationsoptionen nicht verwenden. Weitere Informationen finden Sie unter Amazon ECS Container Agent Configuration im Entwicklerhandbuch zum Amazon Elastic Container Service.

`awslogs`

Gibt den Amazon CloudWatch Logs-Protokollierungstreiber an. Weitere Informationen finden Sie unter Verwenden Sie den awslogs-Protokolltreiber und Amazon CloudWatch Logs-Protokollierungstreiber in der Docker-Dokumentation.

`fluentd`

Gibt den Fluentd-Protokolltreiber an. Weitere Informationen, einschließlich Verwendung und Optionen, finden Sie unter Fluentd-Logging-Treiber in der Docker-Dokumentation.

`gelf`

Gibt den Graylog Extended Format (GELF)-Protokolltreiber an. Weitere Informationen, einschließlich Verwendung und Optionen, finden Sie unter Graylog Extended Format-Logging-Treiber in der Docker-Dokumentation.

`journald`

Gibt den Journald-Protokolltreiber an. Weitere Informationen, einschließlich Verwendung und Optionen, finden Sie unter Journald Logging-Treiber in der Docker-Dokumentation.

`json-file`

Gibt den Treiber für die JSON-Datei an. Weitere Informationen, einschließlich Verwendung und Optionen, finden Sie unter Treiber für die JSON-Dateiprotokollierung in der Docker-Dokumentation.

`splunk`

Gibt den Splunk-Protokolltreiber an. Weitere Informationen, einschließlich Verwendung und Optionen, finden Sie unter Splunk-Logging-Treiber in der Docker-Dokumentation.

`syslog`

Gibt den Syslog-Protokolltreiber an. Weitere Informationen, einschließlich Verwendung und Optionen, finden Sie unter Syslog-Logging-Treiber in der Docker-Dokumentation.

Typ: Zeichenfolge

Erforderlich: Ja

Zulässige Werte: `awslogs` | `fluentd` | `gelf` | `journald` | `json-file` | `splunk` | `syslog`

> **Note**
>
> Wenn Sie einen benutzerdefinierten Treiber haben, der nicht zuvor aufgeführt ist und den Sie mit dem Amazon ECS-Container-Agenten verwenden möchten, können Sie das Amazon ECS-Container-Agent-Projekt, auf dem verfügbar ist, forken GitHub und es so anpassen, dass es mit diesem Treiber funktioniert. Wir möchten Sie bitten, uns eventuelle Änderungswünsche mitzuteilen. Amazon Web Services unterstützt derzeit jedoch keine Anfragen, die modifizierte Kopien dieser Software ausführen.

`options`

Protokollkonfigurationsoptionen, die für den Job an einen Protokolltreiber gesendet werden sollen.

Für diesen Parameter muss Ihre Docker Remote API Version 1.19 oder höher in Ihrer Container-Instance verwenden.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Erforderlich: Nein

`secretOptions`

Ein Objekt, welches das Secret darstellt, der an die Protokollkonfiguration übergeben werden soll. Weitere Informationen finden Sie unter Geben Sie sensible Daten an.

Typ: Objekt-Array

Erforderlich: Nein

`name`

Der Name der Protokolltreiberoption, die im Job festgelegt werden soll.

Typ: Zeichenfolge

Erforderlich: Ja

`valueFrom`

Der Amazon-Ressourcenname (ARN) des Geheimnisses, das in der Protokollkonfiguration des Containers offengelegt werden soll. Die unterstützten Werte sind entweder der

vollständige ARN des Secrets Manager Manager-Geheimnisses oder der vollständige ARN des Parameters im SSM-Parameterspeicher.

> **Note**
>
> Wenn der Parameter SSM Parameter Store in derselben Datei AWS-Region wie die Aufgabe, die Sie starten, vorhanden ist, können Sie entweder den vollständigen ARN oder den Namen des Parameters verwenden. Wenn der Parameter in einer anderen Region existiert, muss der volle ARN angegeben werden.

Typ: Zeichenfolge

Erforderlich: Ja

`memory`

Dieser Parameter ist veraltet. Verwenden Sie ihn stattdessen. *`resourceRequirements`*

Die Anzahl der MiB an Speicher, die für den Job reserviert sind.

Als Beispiel für die Verwendung`resourceRequirements`, wenn Ihre Jobdefinition eine Syntax enthält, die der folgenden ähnelt.

```
"containerProperties": {
  "memory": 512
}
```

Die entsprechende Syntax für die Verwendung `resourceRequirements` lautet wie folgt.

```
"containerProperties": {
  "resourceRequirements": [
    {
      "type": "MEMORY",
      "value": "512"
    }
  ]
}
```

Typ: Ganzzahl

Erforderlich: Ja

`mountPoints`

Die Mountingpunkte für Daten-Volumes in Ihrem Container. Dieser Parameter ordnet zu `Volumes` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--volume` für die docker run zu.

```
"mountPoints": [
                {
                  "sourceVolume": "string",
                  "containerPath": "string",
                  "readOnly": true|false
                }
              ]
```

Typ: Objekt-Array

Erforderlich: Nein

`sourceVolume`

Der Name des zu mountenden Volumes.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `mountPoints` verwendet wird.

`containerPath`

Der Pfad auf dem Container, in den das Host-Volume gemountet werden soll.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `mountPoints` verwendet wird.

`readOnly`

Wenn dieser Wert `true` lautet, verfügt der Container über schreibgeschützten Zugriff auf das Volume. Lautet der Wert `false`, dann verfügt der Container über Schreibzugriff auf das Volume.

Typ: Boolesch

Erforderlich: Nein

Standard: False

`networkConfiguration`

Die Netzwerkkonfiguration für Jobs, die auf Fargate-Ressourcen ausgeführt werden. Jobs, die auf EC2 Ressourcen ausgeführt werden, dürfen diesen Parameter nicht angeben.

```
"networkConfiguration": {
   "assignPublicIp": "string"
}
```

Typ: Objekt-Array

Erforderlich: Nein

`assignPublicIp`

Gibt an, ob die Aufgabe eine öffentliche IP-Adresse hat. Dies ist erforderlich, wenn der Job ausgehenden Netzwerkzugriff benötigt.

Typ: Zeichenfolge

Zulässige Werte: `ENABLED` | `DISABLED`

Erforderlich: Nein

Standard: `DISABLED`

`privileged`

Wenn dieser Parameter den Wert „true“ aufweist, erhält der Container erhöhte Berechtigungen auf der Host-Container-Instance (ähnlich wie der `root`-Benutzer). Dieser Parameter ordnet zu `Privileged` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--privileged` für die docker run zu. Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden. Geben Sie es nicht an und geben Sie es nicht als falsch an.

```
"privileged": true|false
```

Typ: Boolesch

Erforderlich: Nein

`readonlyRootFilesystem`

Wenn dieser Parameter den Wert „true“ aufweist, erhält der Container Lesezugriff auf das Root-Dateisystem. Dieser Parameter ordnet zu `ReadonlyRootfs` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--read-only` für die docker run zu.

```
"readonlyRootFilesystem": true|false
```

Typ: Boolesch

Erforderlich: Nein

`resourceRequirements`

Die Art und Menge einer Ressource, die einem Container zugewiesen werden soll. Zu den unterstützten Ressourcen gehören `GPU`, `MEMORY` und `VCPU`.

```
"resourceRequirements" : [
  {
    "type": "GPU",
    "value": "number"
  }
]
```

Typ: Objekt-Array

Erforderlich: Nein

`type`

Die Art einer Ressource, die einem Container zugewiesen werden soll. Zu den unterstützten Ressourcen gehören `GPU`, `MEMORY` und `VCPU`.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `resourceRequirements` verwendet wird.

`value`

Die Menge der angegebenen Ressource, die für den Container reserviert werden soll. Die Werte variieren abhängig vom angegebenen `type`.

type="GPU"

Die Anzahl der physischen Objekte GPUs , die für den Container reserviert werden sollen. Die Anzahl der für alle Container in einem Job GPUs reservierten Container darf die Anzahl der verfügbaren Container GPUs auf der Rechenressource, auf der der Job gestartet wird, nicht überschreiten.

Typ="SPEICHER"

Die harte Arbeitsspeichergrenze (in MiB), die dem Container bereitgestellt wird. Wenn Ihr Container versucht, das hier angegebene Limit zu überschreiten, wird der Container beendet. Dieser Parameter ist `Memory` im Abschnitt Einen Container erstellen der Docker-Remote-API und der Option `--memory` für die Docker-Ausführung zugeordnet. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen. Dies ist erforderlich, kann jedoch an mehreren Stellen für parallele (MNP)-Aufgaben mit mehreren Knoten angegeben werden. Es muss mindestens einmal für jeden Knoten angegeben werden. Dieser Parameter ist `Memory` im Abschnitt Einen Container erstellen der Docker-Remote-API und der Option `--memory` für die Docker-Ausführung zugeordnet.

> **Note**
>
> Wenn Sie versuchen, Ihre Ressourcennutzung zu maximieren, indem Sie Ihren Jobs so viel Speicher wie möglich für einen bestimmten Instance-Typ zur Verfügung stellen, finden Sie weitere Informationen unterSpeicherverwaltung für Rechenressourcen.

Für Jobs, die auf Fargate-Ressourcen ausgeführt werden, `value` müssen sie einem der unterstützten Werte entsprechen. Außerdem müssen die `VCPU` Werte einer der Werte sein, die für diesen Speicherwert unterstützt werden.

VCPU	MEMORY
0,25 vCPU	512, 1024 und 2048 MiB
0,5 vCPU	1024-4096 MiB in Schritten von 1024 MiB
1 vCPU	2048-8192 MiB in Schritten von 1024 MiB

VCPU	MEMORY
2 vCPU	4096-16384 MiB in Schritten von 1024 MiB
4 vCPU	8192-30720 MiB in Schritten von 1024 MiB
8 vCPU	16384-61440 MiB in Schritten von 4096 MiB
16 vCPU	32768-122880 MiB in Schritten von 8192 MiB

Typ="VCPU“

Die Nummer von v, die für den Job reserviert sind. CPUs Dieser Parameter ist `CpuShares` im Abschnitt Einen Container erstellen der Docker-Remote-API und der Option `--cpu-shares` für die Docker-Ausführung zugeordnet. Jede vCPU entspricht 1.024 CPU-Anteilen. Für Jobs, die auf EC2 Ressourcen ausgeführt werden, müssen Sie mindestens eine vCPU angeben. Dies ist erforderlich, kann aber an mehreren Stellen angegeben werden. Es muss mindestens einmal für jeden Knoten angegeben werden.

Jobs, die auf Fargate-Ressourcen ausgeführt werden, `value` müssen einem der unterstützten Werte entsprechen und die `MEMORY` Werte müssen einem der Werte entsprechen, die für diesen VCPU-Wert unterstützt werden. Die unterstützten Werte sind 0,25, 0,5, 1, 2, 4, 8 und 16.

Das Standardkontingent für die Fargate On-Demand-vCPU-Ressourcenzahl ist 6 V. CPUs Weitere Informationen zu Fargate-Kontingenten finden Sie unter AWS Fargate-Kontingente in der. Allgemeine Amazon Web Services-Referenz

Typ: Zeichenfolge

Erforderlich: Ja, wenn `resourceRequirements` verwendet wird.

`secrets`

Die Geheimnisse für den Job, die als Umgebungsvariablen verfügbar gemacht werden. Weitere Informationen finden Sie unter Geben Sie sensible Daten an.

```
"secrets": [
    {
      "name": "secretName1",
      "valueFrom": "secretArn1"
```

```
    },
    {
      "name": "secretName2",
      "valueFrom": "secretArn2"
    }
    ...
]
```

Typ: Objekt-Array

Erforderlich: Nein

`name`

Der Name der Umgebungsvariablen, die das Geheimnis enthält.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `secrets` verwendet wird.

`valueFrom`

Das Geheimnis, das dem Container exponiert werden muss. Die unterstützten Werte sind entweder der vollständige Amazon Resource Name (ARN) des Secrets Manager Manager-Geheimnisses oder der vollständige ARN des Parameters im SSM Parameter Store.

> **Note**
>
> Wenn der SSM Parameter Store-Parameter in derselben Datei AWS-Region wie der Job, den Sie starten, vorhanden ist, können Sie entweder den vollständigen ARN oder den Namen des Parameters verwenden. Wenn der Parameter in einer anderen Region existiert, muss der volle ARN angegeben werden.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `secrets` verwendet wird.

`ulimits`

Eine Liste der `ulimits`-Werte, die im Container festgelegt werden sollen. Dieser Parameter ordnet zu `Ulimits` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--ulimit` für die docker run zu.

```
"ulimits": [
  {
    "name": string,
    "softLimit": integer,
    "hardLimit": integer
  }
  ...
]
```

Typ: Objekt-Array

Erforderlich: Nein

`name`

Der `type` des `ulimit`-Werts.

Typ: Zeichenfolge

Erforderlich: Ja, wenn `ulimits` verwendet wird.

`hardLimit`

Das harte Limit für den `ulimit`-Typ.

Typ: Ganzzahl

Erforderlich: Ja, wenn `ulimits` verwendet wird.

`softLimit`

Das weiche Limit für den `ulimit`-Typ.

Typ: Ganzzahl

Erforderlich: Ja, wenn `ulimits` verwendet wird.

`user`

Der Benutzername, der im Container verwendet werden soll. Dieser Parameter ordnet zu `User` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--user` für die docker run zu.

```
"user": "string"
```

Typ: Zeichenfolge

Erforderlich: Nein

`vcpus`

Dieser Parameter ist veraltet. Verwenden Sie ihn stattdessen. *`resourceRequirements`*

Die Nummer von v, die für den CPUs Container reserviert ist.

Als Beispiel für die Verwendung`resourceRequirements`, falls Ihre Jobdefinition Zeilen enthält, die der folgenden ähneln:

```
"containerProperties": {
  "vcpus": 2
}
```

Die entsprechende Verwendung von Zeilen `resourceRequirements` lautet wie folgt.

```
"containerProperties": {
  "resourceRequirements": [
    {
      "type": "VCPU",
      "value": "2"
    }
  ]
}
```

Typ: Ganzzahl

Erforderlich: Ja

`volumes`

Wenn Sie eine Auftragsdefinition registrieren, können Sie eine Liste der Volumes angeben, die an den Docker-Daemon in einer Container-Instance übergeben werden sollen. Die folgenden Parameter sind in den Container-Eigenschaften zulässig:

```
"volumes": [
  {
```

```
    "name": "string",
    "host": {
      "sourcePath": "string"
    },
    "efsVolumeConfiguration": {
      "authorizationConfig": {
        "accessPointId": "string",
        "iam": "string"
      },
      "fileSystemId": "string",
      "rootDirectory": "string",
      "transitEncryption": "string",
      "transitEncryptionPort": number
    }
  }
]
```

`name`

Der Name des Volumes. Bis zu 255 Buchstaben (Groß- und Kleinbuchstaben), Ziffern, Bindestriche und Unterstriche sind zulässig. Auf diesen Namen wird im Parameter `sourceVolume` der Containerdefinition `mountPoints` verwiesen.

Typ: Zeichenfolge

Erforderlich: Nein

`host`

Der Inhalt des Parameters `host` bestimmt, ob Ihr Daten-Volume auf der Host-Container-Instance erhalten bleibt und wo es gespeichert wird. Wenn der Parameter `host` leer ist, weist der Docker-Daemon einen Host-Pfad für Ihr Daten-Volume zu. Es kann jedoch nicht garantiert werden, dass die Daten auch dann bestehen bleiben, wenn der ihnen zugeordnete Container nicht mehr läuft.

> **Note**
>
> Dieser Parameter ist nicht auf Aufträge anwendbar, die auf Fargate-Ressourcen ausgeführt werden.

Typ: Objekt

Erforderlich: Nein

`sourcePath`

Der Pfad auf der Host-Container-Instance, die dem Container angezeigt wird. Wenn dieser Parameter leer ist, weist der Docker-Daemon einen Host-Pfad für Sie zu.

Wenn der Parameter `host` den Speicherort `sourcePath` enthält, bleibt das Daten-Volume an der angegebenen Position der Host-Container-Instance erhalten, bis Sie es manuell löschen. Wenn der `sourcePath`-Wert auf der Host-Container-Instance nicht vorhanden ist, wird der Docker-Daemon ihn erstellen. Wenn der Speicherort nicht vorhanden ist, wird der Inhalt des Quellpfadordners exportiert.

Typ: Zeichenfolge

Erforderlich: Nein

`efsVolumeConfiguration`

Dieser Parameter wird angegeben, wenn Sie ein Dateisystem von Amazon Elastic File System für die Aufgabenspeicherung verwenden. Weitere Informationen finden Sie unter Amazon EFS-Volumes.

Typ: Objekt

Erforderlich: Nein

`authorizationConfig`

Die Autorisierungskonfigurationsdetails für das Amazon EFS-Dateisystem.

Typ: Zeichenfolge

Erforderlich: Nein

`accessPointId`

Die Amazon EFS-Zugriffspunkt-ID, die verwendet werden soll. Wenn ein Zugriffspunkt angegeben ist, `EFSVolumeConfiguration` muss der in der angegebene Stammverzeichniswert entweder weggelassen oder auf `/` gesetzt werden. Dadurch wird der Pfad durchgesetzt, der auf dem EFS-Zugriffspunkt festgelegt ist. Wenn ein Zugriffspunkt verwendet wird, muss in `EFSVolumeConfiguration` die Transitverschlüsselung aktiviert sein. Weitere Informationen finden Sie unter Arbeiten mit Amazon EFS-Zugriffspunkten im Amazon Elastic File System-Benutzerhandbuch.

Typ: Zeichenfolge

Erforderlich: Nein

`iam`

Legt fest, ob beim Mounten des Amazon EFS-Dateisystems die in einer Auftragsdefinition definierte AWS Batch Job-IAM-Rolle verwendet werden soll. Wenn diese Option aktiviert ist, muss die Transit-Verschlüsselung in `EFSVolumeConfiguration` aktiviert sein. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert `DISABLED` verwendet. Weitere Informationen finden Sie unter Verwenden Sie Amazon EFS-Zugriffspunkte.

Typ: Zeichenfolge

Zulässige Werte: `ENABLED` | `DISABLED`

Erforderlich: Nein

`fileSystemId`

Die zu verwendende Amazon EFS-Dateisystem-ID.

Typ: Zeichenfolge

Erforderlich: Nein

`rootDirectory`

Das Verzeichnis im Amazon EFS-Dateisystem, das als Stammverzeichnis im Host bereitgestellt werden soll. Wenn dieser Parameter weggelassen wird, wird der Stamm des Amazon-EFS-Volumes verwendet. Wenn Sie dies angeben`/`, hat dies dieselbe Wirkung wie das Weglassen dieses Parameters. Die maximale Länge beträgt 4,096 Zeichen.

> **Important**
>
> Wenn in der ein EFS-Zugriffspunkt angegeben ist`authorizationConfig`, muss der Stammverzeichnisparameter entweder weggelassen oder auf gesetzt werden`/`. Dadurch wird der Pfad durchgesetzt, der auf dem Amazon EFS-Zugriffspunkt festgelegt ist.

Typ: Zeichenfolge

Erforderlich: Nein

`transitEncryption`

Legt fest, ob die Verschlüsselung für Amazon-EFS-Daten während der Übertragung zwischen dem Amazon-ECS-Host und dem Amazon-EFS-Server aktiviert werden soll. Die Transit-Verschlüsselung muss aktiviert sein, wenn die Amazon-EFS-IAM-Autorisierung verwendet wird. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert `DISABLED` verwendet. Weitere Informationen finden Sie unter Verschlüsseln von Daten während der Übertragung im Benutzerhandbuch für Amazon Elastic File System.

Typ: Zeichenfolge

Zulässige Werte: `ENABLED` | `DISABLED`

Erforderlich: Nein

`transitEncryptionPort`

Der zu verwendende Port zum Senden verschlüsselter Daten zwischen dem Amazon-ECS-Host und dem Amazon EFS-Server. Wenn Sie keinen Transit-Verschlüsselungsport angeben, wird die Port-Auswahlstrategie verwendet, die der Amazon EFS-Mount-Helfer verwendet. Dieser Wert muss zwischen 0 und 65.535 liegen. Weitere Informationen finden Sie unter EFS-Mount-Helfer im Benutzerhandbuch für Amazon Elastic File System.

Typ: Ganzzahl

Erforderlich: Nein

Amazon EKS-Eigenschaften

Ein Objekt mit verschiedenen Eigenschaften, die für Amazon-EKS-basierte Aufträge spezifisch sind. Dies darf für Amazon ECS-basierte Jobdefinitionen nicht angegeben werden.

`podProperties`

Die Eigenschaften für die Kubernetes Pod-Ressourcen eines Jobs.

Typ: EksPodProperties Objekt

Erforderlich: Nein

`containers`

Die Eigenschaften des Containers, der im Amazon-EKS-Pod verwendet wird.

Typ: EksContainer Objekt

Erforderlich: Nein

`args`

Eine Reihe von Argumenten für den Einstiegspunkt. Wenn dies nicht angegeben ist, wird der `CMD` des Container-Images verwendet. Dies entspricht dem `args` Mitglied im Entrypoint-Bereich des Pods in. Kubernetes Verweise auf Umgebungsvariablen werden mithilfe der Umgebung des Containers erweitert.

Wenn die referenzierte Umgebungsvariable nicht existiert, wird die Referenz im Befehl nicht geändert. Wenn der Verweis beispielsweise „`$(NAME1)`“ lautet und die `NAME1`-Umgebungsvariable nicht existiert, bleibt die Befehlszeichenfolge „`$(NAME1)`“ erhalten. `$$` wird ersetzt durch `$` und die resultierende Zeichenfolge wird nicht erweitert. Beispielsweise wird `$$(VAR_NAME)` als `$(VAR_NAME)` übergeben, unabhängig davon, ob die Umgebungsvariable `VAR_NAME` vorhanden ist oder nicht. Weitere Informationen finden Sie unter CMD in der Dockerfile-Referenz und Definieren Sie einen Befehl und Argumente für einen Pod in der Dokumentation. Kubernetes

Typ: Zeichenfolgen-Array

Erforderlich: Nein

`command`

Der Einstiegspunkt für den Container. Dies wird nicht in einer Shell ausgeführt. Wenn dies nicht angegeben ist, wird der `ENTRYPOINT` des Container-Images verwendet. Verweise auf Umgebungsvariablen werden mithilfe der Umgebung des Containers erweitert.

Wenn die referenzierte Umgebungsvariable nicht existiert, wird die Referenz im Befehl nicht geändert. Wenn der Verweis beispielsweise „`$(NAME1)`“ lautet und die `NAME1`-Umgebungsvariable nicht existiert, bleibt die Befehlszeichenfolge „`$(NAME1)`“ erhalten. `$$` wird ersetzt durch `$`, und die resultierende Zeichenfolge wird nicht erweitert. Beispielsweise wird `$$(VAR_NAME)` als `$(VAR_NAME)` übergeben, unabhängig davon, ob die Umgebungsvariable `VAR_NAME` vorhanden ist oder nicht. Der Einstiegspunkt kann nicht aktualisiert werden. Weitere Informationen finden Sie unter ENTRYPOINT in der

Dockerfile-Referenz und Definieren Sie einen Befehl und Argumente für einen Container und Entrypoint in der Dokumentation. Kubernetes

Typ: Zeichenfolgen-Array

Erforderlich: Nein

`env`

Die Umgebungsvariablen, die an einen Container übergeben werden.

> **Note**
>
> Umgebungsvariablen können nicht mit `AWS_BATCH` beginnen. Diese Benennungskonvention ist Variablen vorbehalten, die festlegen. AWS Batch

Typ: Array von EksContainerEnvironmentVariable-Objekten

Erforderlich: Nein

`name`

Der Name der Umgebungsvariable.

Typ: Zeichenfolge

Erforderlich: Ja

`value`

Der Wert der Umgebungsvariable.

Typ: Zeichenfolge

Erforderlich: Nein

`image`

Das Docker-Image zum Starten des Containers.

Typ: Zeichenfolge

Erforderlich: Ja

`imagePullPolicy`

Die Image-Pull-Richtlinie für den Container. Unterstützte Werte sind `Always`, `IfNotPresent` und `Never`. Dieser Parameter lautet standardmäßig `IfNotPresent`. Wenn das `:latest`-Tag jedoch angegeben ist, ist es standardmäßig `Always`. Weitere Informationen finden Sie in der KubernetesDokumentation unter Bilder aktualisieren.

Typ: Zeichenfolge

Erforderlich: Nein

`name`

Name des Containers. Wenn der Name nicht angegeben ist, wird der Standardname „`Default`“ verwendet. Jeder Container in einem Pod muss einen eindeutigen Namen haben.

Typ: Zeichenfolge

Erforderlich: Nein

`resources`

Die Art und Menge an Ressourcen, die einem Container zugewiesen werden sollen. Zu den unterstützten Ressourcen gehören `memory`, `cpu` und `nvidia.com/gpu`. Weitere Informationen finden Sie in der KubernetesDokumentation unter Ressourcenverwaltung für Pods und Container.

Typ: EksContainerResourceRequirements Objekt

Erforderlich: Nein

`limits`

Typ und Menge der Ressourcen, die für den Container reserviert werden sollen. Die Werte variieren abhängig vom angegebenen `name`. Ressourcen können entweder mithilfe der `limits`- oder `requests`-Objekte angefordert werden.

memory

Die harte Speichergrenze (in MB) für den Container, unter Verwendung ganzer Zahlen, mit dem Suffix „Mi“. Wenn Ihr Container versucht, das angegebene Limit zu überschreiten, wird der Container beendet. Sie müssen mindestens 4 MB

Arbeitsspeicher für einen Auftrag festlegen. `memory` kann in `limits`, `requests` oder in beiden angegeben werden. Wenn `memory` in beiden Orten angegeben ist, muss der Wert, der in `limits` angegeben ist, dem Wert entsprechen, der in `requests` angegeben ist.

> **Note**
>
> Um Ihre Ressourcenauslastung zu maximieren, stellen Sie Ihren Aufträgen so viel Arbeitsspeicher wie möglich für den spezifischen Instance-Typ zur Verfügung, den Sie verwenden. Um zu erfahren wie dies geht, vgl. Speicherverwaltung für Rechenressourcen.

cpu

Die Nummer CPUs davon ist für den Container reserviert. Werte müssen ein gerade Vielfaches von `0.25` sein. `cpu` kann in `limits`, `requests` oder in beiden angegeben werden. Wenn `cpu` in beiden Orten angegeben ist, muss der Wert, der in `limits` angegeben ist, mindestens so groß sein wie der Wert, der in `requests` angegeben ist.

nvidia.com/gpu

Die Nummer GPUs davon ist für den Container reserviert. Werte müssen eine ganze Zahl sein. `memory` kann in `limits`, `requests` oder in beiden angegeben werden. Wenn `memory` in beiden Orten angegeben ist, muss der Wert, der in `limits` angegeben ist, dem Wert entsprechen, der in `requests` angegeben ist.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Längenbeschränkungen des Wertes: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

`requests`

Typ und Menge der für den Container anzufordernden Ressourcen. Die Werte variieren abhängig vom angegebenen `name`. Ressourcen können entweder mithilfe der `limits`-Objekte oder der `requests`-Objekte angefordert werden.

memory

Die harte Speichergrenze (in MB) für den Container, unter Verwendung ganzer Zahlen, mit dem Suffix „Mi“. Wenn Ihr Container versucht, das angegebene Limit zu überschreiten, wird der Container beendet. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen. `memory` kann in `limits`, `requests` oder in beiden angegeben werden. Wenn `memory` in beiden angegeben ist, muss der Wert, der in `limits` angegeben ist, dem Wert entsprechen, der in `requests` angegeben ist.

> **Note**
>
> Wenn Sie versuchen, Ihre Ressourcennutzung zu maximieren, indem Sie Ihren Jobs so viel Speicher wie möglich für einen bestimmten Instance-Typ zur Verfügung stellen, finden Sie weitere Informationen unterSpeicherverwaltung für Rechenressourcen.

cpu

Die Anzahl CPUs davon ist für den Container reserviert. Werte müssen ein gerade Vielfaches von `0.25` sein. `cpu` kann in `limits`, `requests` oder in beiden angegeben werden. Wenn `cpu` in beiden angegeben ist, muss der Wert, der in `limits` angegeben ist, mindestens so groß sein wie der Wert, der in `requests` angegeben ist.

nvidia.com/gpu

Die Anzahl GPUs davon ist für den Container reserviert. Werte müssen eine ganze Zahl sein. `nvidia.com/gpu` kann in `limits`, `requests` oder in beiden angegeben werden. Wenn `nvidia.com/gpu` in beiden angegeben ist, muss der Wert, der in `limits` angegeben ist, dem Wert entsprechen, der in `requests` angegeben ist.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Längenbeschränkungen des Wertes: Minimale Länge von 1. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

`securityContext`

Der Sicherheitskontext für einen Auftrag. Weitere Informationen finden Sie in der KubernetesDokumentation unter Konfigurieren eines Sicherheitskontextes für einen Pod oder Container.

Typ: EksContainerSecurityContext Objekt

Erforderlich: Nein

`privileged`

Wenn dieser Parameter den Wert `true` aufweist, erhält der Container erhöhte Berechtigungen auf der Host-Container-Instance. Die Ebene der Berechtigungen ähnelt den `root` Benutzerberechtigungen. Der Standardwert ist `false`. Dieser Parameter entspricht der `privileged` Richtlinie in den Sicherheitsrichtlinien für privilegierte Pods in der KubernetesDokumentation.

Typ: Boolesch

Erforderlich: Nein

`readOnlyRootFilesystem`

Wenn dieser Parameter den Wert `true` aufweist, erhält der Container Lesezugriff auf das Root-Dateisystem. Der Standardwert ist `false`. Dieser Parameter ist der `ReadOnlyRootFilesystem` Richtlinie in den Pod-Sicherheitsrichtlinien für Volumes und Dateisysteme in der KubernetesDokumentation zugeordnet.

Typ: Boolesch

Erforderlich: Nein

`runAsGroup`

Wenn dieser Parameter angegeben ist, wird der Container mit der angegebenen Gruppen-ID (`gid`) ausgeführt. Wird dieser Parameter nicht angegeben, wird standardmäßig die Gruppe verwendet, die in den Image-Metadaten angegeben ist. Dieser Parameter ist einer `MustRunAs` Richtlinie in den Pod-Sicherheitsrichtlinien für Benutzer und Gruppen in der KubernetesDokumentation zugeordnet. `RunAsGroup`

Type: Long

Erforderlich: Nein

`runAsNonRoot`

Wenn dieser Parameter angegeben ist, wird der Container als Benutzer mit einem `uid` mit anderen Wert als 0 ausgeführt. Wenn dieser Parameter nicht angegeben ist, wird eine solche Regel durchgesetzt. Dieser Parameter ist einer `MustRunAsNonRoot` Richtlinie in den Pod-Sicherheitsrichtlinien für Benutzer und Gruppen in der KubernetesDokumentation zugeordnet. `RunAsUser`

Type: Long

Erforderlich: Nein

`runAsUser`

Wenn dieser Parameter angegeben ist, wird der Container mit der angegebenen Benutzer-ID (`uid`) ausgeführt. Wird dieser Parameter nicht angegeben, wird standardmäßig der in den Image-Metadaten angegebene Benutzer verwendet. Dieser Parameter ist einer `MustRanAs` Richtlinie in den Pod-Sicherheitsrichtlinien für Benutzer und Gruppen in der KubernetesDokumentation zugeordnet. `RunAsUser`

Type: Long

Erforderlich: Nein

`volumeMounts`

Die Volume-Mounts für einen Container für einen Amazon-EKS-Auftrag. Weitere Informationen zu Volumes und zum Einhängen von Volumes finden Sie in Kubernetes der KubernetesDokumentation unter Volumes.

Typ: Array von EksContainerVolumeMount-Objekten

Erforderlich: Nein

`mountPath`

Der Pfad auf dem Container, in dem das Volume gemountet ist.

Typ: Zeichenfolge

Erforderlich: Nein

`name`

Der Name des zu mountenden Volumes. Dieser muss mit dem Namen eines der Volumes im Pod übereinstimmen.

Typ: Zeichenfolge

Erforderlich: Nein

`readOnly`

Wenn dieser Wert `true` lautet, verfügt der Container über schreibgeschützten Zugriff auf das Volume. Andernfalls kann der Container auf das Volume schreiben. Der Standardwert ist `false`.

Typ: Boolesch

Erforderlich: Nein

`dnsPolicy`

Die DNS-Richtlinie für den Pod. Der Standardwert ist `ClusterFirst`. Wenn der `hostNetwork`-Parameter nicht angegeben ist, ist der Standardwert `ClusterFirstWithHostNet`. `ClusterFirst` gibt an, dass jede DNS-Abfrage, die nicht mit dem konfigurierten Cluster-Domain-Suffix übereinstimmt, an den vom Knoten geerbten Upstream-Nameserver weitergeleitet wird. Wenn `dnsPolicy` in der RegisterJobDefinitionAPI-Operation kein Wert für angegeben wurde, wird weder `dnsPolicy` von noch von DescribeJobDefinitionsDescribeJobsAPI-Vorgängen ein Wert zurückgegeben. Die Pod-Spezifikationseinstellung enthält entweder `ClusterFirst` oder `ClusterFirstWithHostNet`, abhängig vom Wert des `hostNetwork`-Parameters. Weitere Informationen finden Sie in der KubernetesDokumentation zur DNS-Richtlinie von Pod.

Zulässige Werte: `Default` | `ClusterFirst` | `ClusterFirstWithHostNet`

Typ: Zeichenfolge

Erforderlich: Nein

`hostNetwork`

Gibt an, ob der Pod die Netzwerk-IP-Adresse der Hosts verwendet. Der Standardwert ist `true`. Wenn Sie dies auf einstellen, wird das Kubernetes Pod-Netzwerkmodell `false` aktiviert. Die meisten AWS Batch Workloads sind ausschließlich für ausgehende Verbindungen bestimmt und erfordern nicht den Mehraufwand der IP-Zuweisung für jeden Pod

für eingehende Verbindungen. Weitere Informationen finden Sie in der Dokumentation unter Host-Namespaces und Pod-Netzwerke. Kubernetes

Typ: Boolesch

Erforderlich: Nein

`serviceAccountName`

Der Name des Servicekontos, das zum Ausführen des Pods verwendet wird. Weitere Informationen finden Sie unter KubernetesDienstkonten und Konfiguration eines Kubernetes Dienstkontos für die Übernahme einer IAM-Rolle im Amazon EKS-Benutzerhandbuch und Servicekonten für Pods konfigurieren in der KubernetesDokumentation.

Typ: Zeichenfolge

Erforderlich: Nein

`volumes`

Gibt die Volumes für eine Auftragsdefinition an, die Amazon-EKS-Ressourcen verwendet.

Typ: Array von EksVolume-Objekten

Erforderlich: Nein

EmptyDir

Gibt die Konfiguration eines Volumes an Kubernetes`emptyDir`. Ein `emptyDir`-Volume wird zuerst erstellt, wenn ein Pod einem Knoten zugewiesen wird. Es existiert, solange der Pod auf diesem Knoten läuft. Das `emptyDir`-Volume ist zunächst leer. Alle Container im Pod können die Dateien im `emptyDir`-Volume lesen und schreiben. Das `emptyDir`-Volume kann jedoch in jedem Container auf den gleichen oder unterschiedlichen Pfaden gemountet werden. Wenn ein Pod aus irgendeinem Grund von einem Knoten entfernt wird, werden die Daten in `emptyDir` dauerhaft gelöscht. Weitere Informationen finden Sie in der Dokumentation unter EmptyDir. Kubernetes

Typ: EksEmptyDir Objekt

Erforderlich: Nein

Medium

Das Medium zum Speichern des Volumes. Der Standardwert ist eine leere Zeichenfolge, die den Speicher des Knotens verwendet.

""

(Standard) Verwenden Sie den Festplattenspeicher des Knotens.

„Arbeitsspeicher“

Verwenden Sie das `tmpfs`-Volume, das vom RAM des Knotens unterstützt wird. Der Inhalt des Volumes geht verloren, wenn der Knoten neu gestartet wird, und jeder Speicherplatz auf dem Volume wird auf das Speicherlimit des Containers angerechnet.

Typ: Zeichenfolge

Erforderlich: Nein

Größenbeschränkung

Die maximale Größe des Volumes. Standardmäßig ist keine maximale Größe definiert.

Typ: Zeichenfolge

Längenbeschränkungen: Minimale Länge beträgt 1 Zeichen. Maximale Länge beträgt 256 Zeichen.

Erforderlich: Nein

Host-Pfad

Gibt die Konfiguration eines Kubernetes `hostPath` Volumes an. Ein `hostPath`-Volume mountet eine vorhandene Datei oder ein Verzeichnis aus dem Dateisystem des Hostknotens in Ihren Pod. Weitere Informationen finden Sie in der KubernetesDokumentation unter HostPath.

Typ: EksHostPath Objekt

Erforderlich: Nein

path

Der Pfad der Datei oder des Verzeichnisses auf dem Host, das in Containern auf dem Pod gemountet werden soll.

Typ: Zeichenfolge

Erforderlich: Nein

Name

Der Name des Volumes. Der Name muss als DNS-Sub-Domain-Name zulässig sein. Weitere Informationen finden Sie in der Dokumentation unter DNS-Subdomänennamen. Kubernetes

Typ: Zeichenfolge

Erforderlich: Ja

Secret

Gibt die Konfiguration eines Kubernetes `secret` Volumes an. Weitere Informationen finden Sie in der KubernetesDokumentation unter Secret.

Typ: EksSecret Objekt

Erforderlich: Nein

optional

Gibt an, ob das Geheimnis oder die Schlüssel des Geheimnisses definiert werden müssen.

Typ: Boolesch

Erforderlich: Nein

Geheimer Name

Der Name des Secrets. Der Name muss als DNS-Sub-Domain-Name zulässig sein. Weitere Informationen finden Sie in der Dokumentation unter DNS-Subdomänennamen. Kubernetes

Typ: Zeichenfolge

Erforderlich: Ja

Funktionen der Plattform

`platformCapabilities`

Die Plattformfunktionen, die für die Jobdefinition erforderlich sind. Wenn kein Wert angegeben wird, wird standardmäßig der Wert `EC2` verwendet. Für Jobs, die auf Fargate-Ressourcen laufen, `FARGATE` ist angegeben.

> **Note**
>
> Wenn der Job auf Amazon EKS-Ressourcen ausgeführt wird, dürfen Sie nichts angeben`platformCapabilities`.

Typ: Zeichenfolge

Zulässige Werte: `EC2` | `FARGATE`

Erforderlich: Nein

Tags weitergeben

`propagateTags`

Gibt an, ob die Tags aus dem Auftrag oder der Auftragsdefinition an die entsprechende Amazon ECS-Aufgabe weitergegeben werden sollen. Wenn kein Wert angegeben wird, werden die Tags nicht weitergegeben. Tags können nur dann an die Aufgaben weitergegeben werden, wenn die Aufgabe erstellt wird. Bei Tags mit demselben Namen haben Auftrags-Tags Vorrang vor Auftragsdefinitions-Tags. Wenn die Gesamtzahl der kombinierten Tags aus dem Job und der Auftragsdefinition über 50 liegt, wird der Job in den `FAILED` Status verschoben.

> **Note**
>
> Wenn der Job auf Amazon EKS-Ressourcen ausgeführt wird, dürfen Sie nichts angeben`propagateTags`.

Typ: Boolesch

Erforderlich: Nein

Eigenschaften des Knotens

`nodeProperties`

Wenn Sie eine parallel Auftragsdefinition mit mehreren Knoten registrieren, müssen Sie eine Liste von Knoteneigenschaften angeben. Diese Knoteneigenschaften definieren die

Anzahl der Knoten, die in Ihrem Job verwendet werden sollen, den Hauptknotenindex und die verschiedenen zu verwendenden Knotenbereiche. Wenn der Job auf Fargate-Ressourcen ausgeführt wird, können Sie keine Angaben machen. `nodeProperties` Nutzen Sie stattdessen `containerProperties`. Die folgenden Knoteneigenschaften sind in einer Auftragsdefinition zulässig. Weitere Informationen finden Sie unter parallel Jobs mit mehreren Knoten.

> **Note**
>
> Wenn der Job auf Amazon EKS-Ressourcen ausgeführt wird, dürfen Sie nichts angeben`nodeProperties`.

Typ: NodeProperties Objekt

Erforderlich: Nein

`mainNode`

Gibt den Knotenindex für den Hauptknoten eines parallelen Auftrags mit mehreren Knoten an. Dieser Knotenindexwert muss kleiner als die Anzahl der Knoten sein.

Typ: Ganzzahl

Erforderlich: Ja

`numNodes`

Die Anzahl der Knoten, die mit einem parallelen Auftrag mit mehreren Knoten verknüpft sind.

Typ: Ganzzahl

Erforderlich: Ja

`nodeRangeProperties`

Eine Liste von Knotenbereichen und deren Eigenschaften, die mit einem parallelen Auftrag mit mehreren Knoten verknüpft sind.

> **Note**
>
> Eine Knotengruppe ist eine identische Gruppe von Job-Knoten, die alle dieselben Container-Eigenschaften haben. Sie können AWS Batch damit bis zu fünf verschiedene Knotengruppen für jeden Job angeben.

Typ: Array von NodeRangeProperty-Objekten

Erforderlich: Ja

`targetNodes`

Der Bereich der Knoten, die Knotenindexwerte verwenden. Ein Bereich von `0:3` zeigt Knoten mit Indexwerten von `0` bis `3` an. Wenn der Wert für den Startbereich weggelassen wird (`:n`), wird 0 verwendet, um den Bereich zu starten. Wenn der Endbereichswert ausgelassen wird (`n:`), wird für den Endbereich der höchstmögliche Knotenindex verwendet wird. Ihre gesamten Knotenbereiche müssen alle Knoten berücksichtigen (`0:n`). Sie können Knotenbereiche verschachteln, zum Beispiel `0:10` und`4:5`. In diesem Fall überschreiben die `4:5` Bereichseigenschaften die `0:10` Eigenschaften.

Typ: Zeichenfolge

Erforderlich: Nein

`container`

Die Container-Details für den Knotenbereich. Weitere Informationen finden Sie unter Eigenschaften des Containers.

Typ: ContainerProperties Objekt

Erforderlich: Nein

Strategie erneut versuchen

`retryStrategy`

Wenn Sie eine Auftragsdefinition registrieren, können Sie optional eine Wiederholungsstrategie für fehlgeschlagene Aufträge angeben, die mit dieser Auftragsdefinition übermittelt werden. Jede Wiederholungsstrategie, die während eines SubmitJobVorgangs angegeben wird, hat Vorrang vor der hier definierten Wiederholungsstrategie. Standardmäßig erfolgt ein Ausführungsversuch pro Auftrag. Wenn Sie mehr als einen Versuch angeben, wird der Job erneut versucht, falls er fehlschlägt. Beispiele für einen fehlgeschlagenen Versuch sind, dass der Job einen Exit-Code ungleich Null zurückgibt oder die Container-Instance beendet wird. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.

Typ: RetryStrategy Objekt

Erforderlich: Nein

`attempts`

Gibt die Zahl der Wiederholungsversuche an, um einen Auftrag in den Status `RUNNABLE` zu verschieben. Sie können zwischen einem und zehn Versuche angeben. Wenn `attempts` größer als 1 ist, wird der Auftrag bei einem auftretenden Fehler entsprechend oft wiederholt, bis er in den Status `RUNNABLE` wechselt.

```
"attempts": integer
```

Typ: Ganzzahl

Erforderlich: Nein

`evaluateOnExit`

Array mit bis zu 5 Objekten, die Bedingungen angeben, unter denen der Job wiederholt wird oder fehlschlägt. Wenn dieser Parameter angegeben wird, muss der Parameter `attempts` ebenfalls angegeben werden. Wenn angegeben `evaluateOnExit` ist, aber keiner der Einträge zutrifft, wird der Job erneut versucht.

```
"evaluateOnExit": [
   {
      "action": "string",
      "onExitCode": "string",
      "onReason": "string",
      "onStatusReason": "string"
   }
]
```

Typ: Array von EvaluateOnExit-Objekten

Erforderlich: Nein

`action`

Gibt die Aktion an, die durchgeführt werden soll, wenn alle angegebenen Bedingungen (`onStatusReason`, `onReason` und `onExitCode`) erfüllt sind. Bei den Werten muss die Groß- und Kleinschreibung beachtet werden.

Typ: Zeichenfolge

Erforderlich: Ja

Zulässige Werte: `RETRY` | `EXIT`

`onExitCode`

Enthält ein Glob-Muster, das mit der Dezimaldarstellung von verglichen werden soll`ExitCode`, die für einen Job zurückgegeben wurde. Jedes Muster kann bis zu 512 Zeichen lang sein. Es kann nur Zahlen enthalten. Es darf keine Buchstaben oder Sonderzeichen enthalten. Es kann optional mit einem Sternchen (*) enden, so dass nur der Anfang der Zeichenfolge genau übereinstimmen muss.

Typ: Zeichenfolge

Erforderlich: Nein

`onReason`

Enthält ein Glob-Muster, das mit dem verglichen werden soll`Reason`, das für einen Job zurückgegeben wurde. Jedes Muster kann bis zu 512 Zeichen lang sein. Es kann Buchstaben, Zahlen, Punkte (.), Doppelpunkte (:) und Leerzeichen (Leerzeichen, Tabulatoren) enthalten. Es kann optional mit einem Sternchen (*) enden, so dass nur der Anfang der Zeichenfolge genau übereinstimmen muss.

Typ: Zeichenfolge

Erforderlich: Nein

`onStatusReason`

Enthält ein Glob-Muster, das mit dem verglichen werden soll`StatusReason`, das für einen Job zurückgegeben wurde. Jedes Muster kann bis zu 512 Zeichen lang sein. Es kann Buchstaben, Zahlen, Punkte (.), Doppelpunkte (:) und Leerzeichen (Leerzeichen, Tabulatoren) enthalten. Es kann optional mit einem Sternchen (*) enden, so dass nur der Anfang der Zeichenfolge genau übereinstimmen muss.

Typ: Zeichenfolge

Erforderlich: Nein

Priorität bei der Planung

`schedulingPriority`

Die Planungspriorität für Jobs, die mit dieser Jobdefinition eingereicht werden. Dies betrifft nur Jobs in Auftragswarteschlangen mit einer Fair-Share-Richtlinie. Aufgaben mit einer höheren Planungspriorität werden vor Aufgaben mit einer niedrigeren Planungspriorität geplant.

Der unterstützte Mindestwert ist 0 und der unterstützte Höchstwert ist 9999.

Typ: Ganzzahl

Erforderlich: Nein

Tags

`tags`

Schlüssel-Wert-Paar-Tags, die der Jobdefinition zugeordnet werden sollen. Weitere Informationen finden Sie unter Kennzeichnen Sie Ihre AWS Batch Ressourcen.

Typ: Abbildung einer Zeichenfolge auf eine Zeichenfolge

Erforderlich: Nein

Timeout

`timeout`

Sie können eine Timeout-Dauer für Ihre Jobs konfigurieren, sodass der Job AWS Batch beendet wird, wenn er länger läuft. Weitere Informationen finden Sie unter Timeouts bei der Job. Wenn ein Job aufgrund eines Timeouts beendet wird, wird er nicht erneut versucht. Jede Timeout-Konfiguration, die während eines SubmitJobVorgangs angegeben wird, hat Vorrang vor der hier definierten Timeout-Konfiguration. Weitere Informationen finden Sie unter Timeouts bei der Job.

Typ: JobTimeout Objekt

Erforderlich: Nein

`attemptDurationSeconds`

Die Zeitdauer in Sekunden (gemessen anhand des `startedAt` Zeitstempels des Auftragsversuchs) nach AWS Batch Beendigung nicht abgeschlossener Jobs. Der Mindestwert für die Zeitüberschreitung beträgt 60 Sekunden.

Bei Array-Aufträgen gilt das Timeout für die untergeordneten Aufträge, nicht für den übergeordneten Array-Auftrag.

Bei parallel Aufträgen mit mehreren Knoten (MNP) gilt das Timeout für den gesamten Auftrag, nicht für die einzelnen Knoten.

Typ: Ganzzahl

Erforderlich: Nein

Erstellen Sie Jobdefinitionen mit EcsProperties

Mithilfe `EcsProperties`von AWS Batch Jobdefinitionen können Sie Hardware, Sensoren, 3D-Umgebungen und andere Simulationen in separaten Containern modellieren. Sie können diese Funktion verwenden, um Ihre Workload-Komponenten logisch zu organisieren und sie von der Hauptanwendung zu trennen. Diese Funktion kann mit Amazon Elastic Container Service (AWS Batch Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) und verwendet werden. AWS Fargate

`ContainerProperties`im Vergleich zu Jobdefinitionen **`EcsProperties`**

Sie können wählen, ob Sie `EcsProperties`Jobdefinitionen verwenden möchten, je nach Anwendungsfall. `ContainerProperties` Auf hoher Ebene `EcsProperties` ähnelt das Ausführen von AWS Batch Jobs mit dem Ausführen von Jobs mit einem`ContainerProperties`.

Die alte Auftragsdefinitionsstruktur, die verwendet wird`ContainerProperties`, wird weiterhin unterstützt. Wenn Sie derzeit Workflows haben, die diese Struktur verwenden, können Sie sie weiterhin ausführen.

Der Hauptunterschied besteht darin, dass der Auftragsdefinition ein neues Objekt hinzugefügt wurde, um die Basisdefinitionen `EcsProperties` zu berücksichtigen.

Eine Jobdefinition, die `ContainerProperties` auf Amazon ECS und Fargate verwendet wird, hat beispielsweise die folgende Struktur:

```
{
   "containerProperties": {
     ...
     "image": "my_ecr_image1",
     ...
  },
...
}
```

Eine Jobdefinition, die `EcsProperties` auf Amazon ECS und Fargate verwendet wird, hat die folgende Struktur:

```
{
  "ecsProperties": {
    "taskProperties": [{
      "containers": [
        {
          ...
          "image": "my_ecr_image1",
          ...
        },
        {
          ...
          "image": "my_ecr_image2",
          ...
        },
```

Allgemeine Änderungen an der AWS Batch APIs

Im Folgenden werden einige der wichtigsten Unterschiede bei der Verwendung der `EcsProperties` und der `EcsProperties` API-Datentypen näher beschrieben:

- Viele der darin verwendeten Parameter befinden `ContainerProperties` sich in`TaskContainerProperties`. Einige Beispiele umfassen`command`,`image`,`privileged`,`secrets`, und`users`. Sie sind alle darin zu finden TaskContainerProperties.
- Einige der `TaskContainerProperties` Parameter haben in der alten Struktur keine funktionalen Entsprechungen. Einige Beispiele umfassen,,`dependsOn`, `essential` `nameipcMode`, und. `pidMode` Weitere Informationen erhalten Sie unter EcsTaskDetails und TaskContainerProperties.

Außerdem gibt es in der Struktur für einige `ContainerProperties` Parameter keine Entsprechungen oder Anwendungen. `EcsProperties` In `container` wurde durch ersetzt `taskProperties`, `containers` sodass das neue Objekt bis zu zehn Elemente aufnehmen kann. Weitere Informationen finden Sie unter:ContainerProperties und:containersRegisterJobDefinition. EcsTaskProperties

- `taskRoleArn`ist funktionell `jobRoleArn` äquivalent zu. Weitere Informationen finden Sie unter EcsTaskProperties: taskRoleArn und ContainerProperties: jobRoleArn.
- Sie können einen (1) bis zehn (10) Container in die `EcsProperties` Struktur aufnehmen. Weitere Informationen finden Sie unter:ContainerEcsTaskProperties.
- Die Objekte `taskProperties` und instanceTypes sind Arrays, akzeptieren aber derzeit nur ein Element. Zum Beispiel :TaskProperties EcsProperties und:InstanceTypes. NodeRangeProperty

Jobdefinitionen mit mehreren Containern für Amazon ECS

Um der Multi-Container-Struktur für Amazon ECS Rechnung zu tragen, sind einige API-Datentypen unterschiedlich. Zum Beispiel

- `ecsProperties`ist dieselbe Ebene wie `containerProperties` in der Einzelcontainer-Definition. Weitere Informationen finden Sie unter EcsProperties in der AWS Batch -API-Referenz.
- `taskProperties`enthält die für die Amazon ECS-Aufgabe definierten Eigenschaften. Weitere Informationen finden Sie unter EcsProperties in der AWS Batch -API-Referenz.
- `containers`enthält ähnliche Informationen wie `containerProperties` in der Einzelcontainer-Definition. Der Hauptunterschied besteht darin, dass Sie bis zu zehn Container definieren `containers` können. Weitere Informationen finden Sie unter ECSTaskProperties:Containers im AWS Batch API-Referenzhandbuch.
- `essential`Der Parameter gibt an, wie sich der Container auf den Job auswirkt. Alle wichtigen Container müssen erfolgreich abgeschlossen werden (beenden als 0), damit der Job fortgesetzt werden kann. Wenn ein Container, der als essenziell markiert ist, ausfällt (mit einem Wert ungleich 0 beendet wird), schlägt der Job fehl.

  Der Standardwert ist `true` und mindestens ein Container muss als markiert sein. `essential` Weitere Informationen finden Sie unter `essential` in der AWS Batch -API-Referenz.
- Mit dem `dependsOn`Parameter können Sie eine Liste von Container-Abhängigkeiten definieren. Weitere Informationen finden Sie unter `dependsOn` in der AWS Batch -API-Referenz.

> **Note**
>
> Die Komplexität der `dependsOn` Liste und die zugehörige Container-Laufzeit können sich auf die Startzeit Ihres Jobs auswirken. Wenn die Ausführung der Abhängigkeiten lange dauert, bleibt der Job so lange in einem `STARTING` Zustand, bis sie abgeschlossen sind.

Weitere Informationen zur UND-Struktur finden Sie unter RegisterJobDefinitionAnforderungssyntax für ecsProperties. `ecsProperties`

Jobdefinitionen mit mehreren Containern für Amazon EKS

Um der Multi-Container-Struktur für Amazon EKS Rechnung zu tragen, sind einige API-Datentypen unterschiedlich. Zum Beispiel

- `name`ist ein eindeutiger Bezeichner für den Container. Dieses Objekt ist nicht für einen einzelnen Container erforderlich, ist jedoch erforderlich, wenn mehrere Container in einem Pod definiert werden. Wenn es `name` nicht für einzelne Container definiert ist, wird der Standardname`default`, angewendet.
- `initContainers`sind innerhalb des eksPodPropertiesDatentyps definiert. Sie werden vor Anwendungscontainern ausgeführt, laufen immer bis zum Abschluss und müssen erfolgreich abgeschlossen werden, bevor der nächste Container gestartet wird.

  Diese Container sind beim Amazon EKS Connector-Agenten registriert und speichern die Registrierungsinformationen im Backend-Datenspeicher von Amazon Elastic Kubernetes Service. Das `initContainers` Objekt kann bis zu zehn (10) Elemente aufnehmen. Weitere Informationen finden Sie unter Init Containers im Kubernetes Dokumentation.

  > **Note**
  >
  > Das `initContainers` Objekt kann sich auf die Startzeit Ihres Jobs auswirken. Wenn die Ausführung lange `initContainers` dauert, bleibt der Job so lange erhalten, `STARTING` bis sie abgeschlossen sind.

- `shareProcessNamespace`gibt an, ob sich die Container im Pod denselben Prozess-Namespace teilen können. Die Standardwerte sind. `false` Diese Einstellung `true` soll es Containern

ermöglichen, Prozesse in anderen Containern, die sich im selben Pod befinden, zu sehen und zu signalisieren.

- Jeder Container ist wichtig. Alle Container müssen erfolgreich abgeschlossen werden (beenden als 0), damit der Job erfolgreich ist. Wenn ein Container ausfällt (mit einem anderen Wert als 0 beendet wird), schlägt der Job fehl.

Weitere Informationen zur UND-Struktur finden Sie unter RegisterJobDefinitionAnforderungssyntax für eksProperties. `eksProperties`

Referenz: AWS Batch Jobszenarien mit EcsProperties

Um zu veranschaulichen, wie AWS Batch Jobdefinitionen, die Sie verwenden, nach Ihren Bedürfnissen strukturiert werden `EcsProperties` können, werden in diesem Thema die folgenden `RegisterJobDefinition`Payloads vorgestellt. Sie können diese Beispiele in eine Datei kopieren, sie an Ihre Bedürfnisse anpassen und dann mit AWS Command Line Interface (AWS CLI) aufrufen`RegisterJobDefinition`.

AWS Batch Job für Amazon ECS bei Amazon EC2

Im Folgenden finden Sie ein Beispiel für einen AWS Batch Job für Amazon Elastic Container Service auf Amazon Elastic Compute Cloud:

```
{
    "jobDefinitionName": "multicontainer-ecs-ec2",
    "type": "container",
    "ecsProperties": {
        "taskProperties": [
          {
            "containers": [
              {
                "name": "c1",
                "essential": false,
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
```

```
                },
                {
                  "type": "MEMORY",
                  "value": "4096"
                }
              ]
            },
            {
              "name": "c2",
              "essential": false,
              "command": [
                "echo",
                "hello world"
              ],
              "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
              "resourceRequirements": [
                {
                  "type": "VCPU",
                  "value": "2"
                },
                {
                  "type": "MEMORY",
                  "value": "4096"
                }
              ]
            },
            {
              "name": "c3",
              "essential": true,
              "command": [
                "echo",
                "hello world"
              ],
              "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
              "firelensConfiguration": {
                "type": "fluentbit",
                "options": {
                  "enable-ecs-log-metadata": "true"
                }
               },
              "resourceRequirements": [
                {
                  "type": "VCPU",
                  "value": "6"
```

```
                  },
                  {
                    "type": "MEMORY",
                    "value": "12288"
                  }
                ]
              }
            ]
          }
        ]
  }
}
```

AWS Batch Job für Amazon ECS auf Fargate

Im Folgenden finden Sie ein Beispiel für einen AWS Batch Job für Amazon Elastic Container Service am AWS Fargate:

```
{
    "jobDefinitionName": "multicontainer-ecs-fargate",
    "type": "container",
    "platformCapabilities": [
        "FARGATE"
    ],
    "ecsProperties": {
        "taskProperties": [
          {
            "containers": [
              {
                "name": "c1",
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
```

```
                }
              ]
            },
            {
              "name": "c2",
              "essential": true,
              "command": [
                "echo",
                "hello world"
              ],
              "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
              "resourceRequirements": [
                {
                  "type": "VCPU",
                  "value": "6"
                },
                {
                  "type": "MEMORY",
                  "value": "12288"
                }
              ]
            }
          ],
          "executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
        }
      ]
  }
}
```

AWS Batch Job für Amazon EKS

Im Folgenden finden Sie ein Beispiel für einen AWS Batch Job für Amazon Elastic Kubernetes Service:

```
{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {
          "name": "init-container",
```

```
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo"
          ],
          "args": [
            "hello world"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
        {
          "name": "init-container-2",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo",
            "my second init container"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ],
      "containers": [
        {
          "name": "c1",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo world"
         ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
        {
```

```
          "name": "sleep-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "20"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ]
    }
  }
}
```

AWS Batch MNP-Job mit mehreren Containern pro Knoten

Im Folgenden finden Sie ein Beispiel für einen AWS Batch Paralleljob (MNP) mit mehreren Knoten und mehreren Containern pro Knoten:

```
{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {
          "taskProperties": [
            {
              "containers": [
                {
                  "name": "range05-c1",
                  "command": [
                    "echo",
                    "hello world"
                  ],
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
```

```
                    "resourceRequirements": [
                      {
                        "type": "VCPU",
                        "value": "2"
                      },
                      {
                        "type": "MEMORY",
                        "value": "4096"
                      }
                    ]
                  },
                  {
                    "name": "range05-c2",
                    "command": [
                      "echo",
                      "hello world"
                    ],
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                    "resourceRequirements": [
                      {
                        "type": "VCPU",
                        "value": "2"
                      },
                      {
                        "type": "MEMORY",
                        "value": "4096"
                      }
                    ]
                  }
                ]
              }
            ]
          }
        }
      ]
    }
}
```

Verwenden Sie den awslogs-Protokolltreiber

AWS Batch Aktiviert standardmäßig den `awslogs` Protokolltreiber, Protokollinformationen an Logs zu CloudWatch senden. Mit dieser Funktion können Sie verschiedene Logs aus Ihren Containern

an einem zentralen Ort anzeigen und so verhindern, dass Ihre Container-Logs Speicherplatz auf Ihren Container-Instances belegen. Dieses Thema hilft Ihnen bei der Konfiguration des `awslogs` Protokolltreibers in Ihren Jobdefinitionen.

> **Note**
>
> In der AWS Batch Konsole können Sie den `awslogs` Protokolltreiber im Abschnitt Konfiguration der Protokollierung konfigurieren, wenn Sie eine Auftragsdefinition erstellen.

> **Note**
>
> Die Art der Informationen, die von den Containern in Ihrem Job protokolliert werden, hängt hauptsächlich von deren `ENTRYPOINT` Befehl ab. Standardmäßig zeigen die erfassten Protokolle die Befehlsausgabe, die Sie normalerweise in einem interaktiven Terminal sehen, wenn Sie den Container lokal ausgeführt haben, also die I/O-Streams `STDOUT` und `STDERR` I/O-Streams. Der `awslogs` Protokolltreiber übergibt diese Protokolle einfach von Docker an CloudWatch Logs. Weitere Informationen dazu, wie Docker-Protokolle verarbeitet werden, einschließlich alternativer Möglichkeiten zum Erfassen unterschiedlicher Dateidaten oder Streams, finden Sie unter Anzeigen von Protokollen für einen Container oder Service in der Docker-Dokumentation.

Informationen zum Senden von Systemprotokollen von Ihren Container-Instances an CloudWatch Logs finden Sie unterVerwenden von CloudWatch Protokollen mit AWS Batch. Weitere Informationen zu CloudWatch Protokollen finden Sie unter Überwachung von Protokolldateien und CloudWatch Protokollkontingenten im Amazon CloudWatch Logs-Benutzerhandbuch.

awslogs-Protokolltreiberoptionen im Datentyp AWS Batch JobDefiniton

Der `awslogs` Protokolltreiber unterstützt die folgenden Optionen in AWS Batch Jobdefinitionen. Weitere Informationen finden Sie unter Treiber für die CloudWatch Protokollierung von Protokollen in der Docker-Dokumentation.

`awslogs-region`

Erforderlich: Nein

Geben Sie die Region an, in die der `awslogs` Protokolltreiber Ihre Docker-Logs senden soll. Standardmäßig ist die Region, die verwendet wird, dieselbe wie die für den Job verwendete. In Logs können Sie wählen, ob Sie all Ihre Logs von Jobs in verschiedenen Regionen an eine einzige Region senden möchten. CloudWatch Auf diese Weise können sie alle von einem Ort aus sichtbar sein. Alternativ können Sie sie nach Regionen trennen, um einen detaillierteren Ansatz zu erreichen. Wenn Sie diese Option wählen, stellen Sie jedoch sicher, dass die angegebenen Protokollgruppen in der von Ihnen angegebenen Region vorhanden sind.

`awslogs-group`

Erforderlich: Optional

Mit der `awslogs-group` Option können Sie die Protokollgruppe angeben, an die der `awslogs` Protokolltreiber seine Protokollstreams sendet. Wenn dies nicht angegeben `aws/batch/job` ist, wird verwendet.

`awslogs-stream-prefix`

Erforderlich: Optional

Mit der `awslogs-stream-prefix` Option können Sie einen Protokollstream mit dem angegebenen Präfix und der Amazon ECS-Aufgaben-ID des AWS Batch Jobs verknüpfen, zu dem der Container gehört. Wenn Sie einen Präfix mit dieser Option angeben, weist der Protokoll-Stream das folgende Format auf:

```
prefix-name/default/ecs-task-id
```

`awslogs-datetime-format`

Erforderlich: Nein

Diese Option definiert einen mehrzeiliges Startmuster im `strftime`-Format von Python. Eine Protokollnachricht besteht aus einer Zeile, die dem Muster entspricht, und allen folgenden Zeilen, die nicht dem Muster entsprechen. Damit ist die zugeordnete Zeile das Trennzeichen zwischen Protokollnachrichten.

Ein Beispiel für einen Anwendungsfall für die Nutzung dieses Formats ist die Ausgabenanalyse, z. B. einen Stack-Dump, der andernfalls möglicherweise mehrere Einträge protokolliert. Mit dem richtigen Muster kann es in nur einem Eintrag erfasst werden.

Weitere Informationen finden Sie unter awslogs-datetime-format.

Diese Option hat immer Vorrang, wenn sowohl `awslogs-datetime-format` als auch `awslogs-multiline-pattern` konfiguriert sind.

> **Note**
>
> Bei der mehrzeiligen Protokollierung wird eine regelmäßige Ausdrucksanalyse und die Übereinstimmung aller Protokollmeldungen ausgeführt. Dies kann negative Auswirkungen auf die Leistung der Protokollierung haben.

`awslogs-multiline-pattern`

Erforderlich: Nein

Diese Option definiert ein mehrzeiliges Startmuster mit einem regulären Ausdruck. Eine Protokollnachricht besteht aus einer Zeile, die dem Muster entspricht, und allen folgenden Zeilen, die nicht dem Muster entsprechen. Somit ist die übereinstimmende Zeile das Trennzeichen zwischen Protokollnachrichten.

Weitere Informationen finden Sie awslogs-multiline-patternin der Docker-Dokumentation.

Diese Option wird ignoriert, wenn `awslogs-datetime-format` ebenfalls konfiguriert ist.

> **Note**
>
> Bei der mehrzeiligen Protokollierung wird eine regelmäßige Ausdrucksanalyse und die Übereinstimmung aller Protokollmeldungen ausgeführt. Dies kann negative Auswirkungen auf die Leistung der Protokollierung haben.

`awslogs-create-group`

Erforderlich: Nein

Geben Sie an, ob die Protokollgruppe automatisch erstellt werden soll. Wenn diese Option nicht angegeben ist, gilt standardmäßig `false`.

> **Warning**
>
> Diese Option wird nicht empfohlen. Es wird empfohlen, die Protokollgruppe im Voraus mithilfe der CloudWatch CreateLogGroupProtokoll-API-Aktion zu erstellen, da jeder Job

versucht, die Protokollgruppe zu erstellen, wodurch die Wahrscheinlichkeit steigt, dass der Job fehlschlägt.

> **Note**
>
> Die IAM-Richtlinie für Ihre Ausführungsrolle muss die `logs:CreateLogGroup` entsprechende Genehmigung enthalten, bevor Sie versuchen, sie zu verwendenawslogs-create-group.

Geben Sie in Ihrer Jobdefinition eine Protokollkonfiguration an

AWS Batch Aktiviert standardmäßig den `awslogs` Protokolltreiber. In diesem Abschnitt wird beschrieben, wie Sie die `awslogs` Protokollkonfiguration für einen Job anpassen. Weitere Informationen finden Sie unter Erstellen Sie eine Auftragsdefinition mit einem Knoten .

In den folgenden JSON-Snippets für die Protokollkonfiguration ist für jeden `logConfiguration` Job ein Objekt angegeben. Einer ist für einen WordPress Job, der Protokolle an eine Protokollgruppe namens sendetawslogs-wordpress, und ein anderer ist für einen MySQL-Container, der Protokolle an eine Protokollgruppe namens sendetawslogs-mysql. Beide Container verwenden den Protokoll-Stream-Präfix `awslogs-example`.

```
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "awslogs-wordpress",
        "awslogs-stream-prefix": "awslogs-example"
    }
}
```

```
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "awslogs-mysql",
        "awslogs-stream-prefix": "awslogs-example"
    }
}
```

In der AWS Batch Konsole wird die Protokollkonfiguration für die `wordpress` Auftragsdefinition wie in der folgenden Abbildung dargestellt angegeben.

Nachdem Sie eine Aufgabendefinition beim `awslogs` Protokolltreiber in einer Protokollkonfiguration für die Auftragsdefinition registriert haben, können Sie einen Job mit dieser Auftragsdefinition einreichen, um mit dem Senden von Protokollen an Logs zu CloudWatch beginnen. Weitere Informationen finden Sie unter Tutorial: Einen Job einreichen.

Geben Sie sensible Daten an

Mit können Sie vertrauliche Daten in Ihre Jobs einfügen AWS Batch, indem Sie Ihre sensiblen Daten entweder in AWS Secrets Manager Secrets- oder AWS Systems Manager Parameter Store-Parametern speichern und dann in Ihrer Jobdefinition darauf verweisen.

Geheimnisse können einem Job auf folgende Weise zugänglich gemacht werden:

- Verwenden Sie den `secrets` Jobdefinitionsparameter, um sensible Daten als Umgebungsvariablen in Ihre Container einzufügen.
- Verwenden Sie den Jobdefinitionsparameter, um in der Protokollkonfiguration eines Jobs auf vertrauliche Informationen zu verweisen. `secretOptions`

Themen

- Geben Sie sensible Daten mit Secrets Manager an
- Geben Sie vertrauliche Daten mit Systems Manager Parameter Store an

Geben Sie sensible Daten mit Secrets Manager an

Mit können Sie sensible Daten in Ihre Jobs einfügen AWS Batch, indem Sie Ihre sensiblen Daten geheim speichern und dann in AWS Secrets Manager Ihrer Jobdefinition darauf verweisen. Sensible Daten, die in Secrets Manager Secrets gespeichert sind, können einem Job als Umgebungsvariablen oder als Teil der Protokollkonfiguration zugänglich gemacht werden.

Wenn Sie ein Secret als Umgebungsvariable einfügen, können Sie einen JSON-Schlüssel oder eine Version eines einzufügenden Secrets angeben. Dieser Prozess hilft Ihnen dabei, die sensiblen Daten zu kontrollieren, die für Ihren Job offengelegt werden. Weitere Informationen zur geheimen Versionsverwaltung finden Sie unter Schlüsselbegriffe und -konzepte für AWS Secrets Manager im AWS Secrets Manager -Benutzerhandbuch.

Überlegungen bei der Angabe sensibler Daten mit Secrets Manager

Folgendes sollte bei der Verwendung von Secrets Manager zur Angabe sensibler Daten für Jobs berücksichtigt werden.

- Um ein Geheimnis mithilfe eines bestimmten JSON-Schlüssels oder einer bestimmten Version eines Geheimnisses einzufügen, muss auf der Container-Instance in Ihrer Rechenumgebung Version 1.37.0 oder höher des Amazon ECS-Container-Agenten installiert sein. Wir empfehlen jedoch, die neueste Container-Agent-Version zu verwenden. Informationen zur Überprüfung Ihrer Agentenversion und zur Aktualisierung auf die neueste Version finden Sie unter Aktualisieren des Amazon ECS-Container-Agenten im Amazon Elastic Container Service Developer Guide.

  Um den vollständigen Inhalt eines Secrets als Umgebungsvariable oder um ein Secret in eine Protokollkonfiguration einzufügen, muss Ihre Container-Instance über Version 1.23.0 oder höher des Container-Agenten verfügen.
- Es werden nur Geheimnisse unterstützt, die Textdaten speichern, also Geheimnisse, die mit dem `SecretString` CreateSecretAPI-Parameter erstellt wurden. Geheimnisse, die Binärdaten speichern, bei denen es sich um Geheimnisse handelt, die mit dem `SecretBinary` CreateSecretAPI-Parameter erstellt wurden, werden nicht unterstützt.

- Wenn Sie eine Jobdefinition verwenden, die Secrets Manager-Secrets referenziert, um sensible Daten für Ihre Jobs abzurufen, und Sie auch Schnittstellen-VPC-Endpunkte verwenden, müssen Sie die Schnittstellen-VPC-Endpunkte für Secrets Manager erstellen. Weitere Informationen finden Sie unter Verwenden von Secrets Manager mit VPC-Endpunkten im AWS Secrets Manager - Benutzerhandbuch.
- Sensible Daten werden in Ihren Job eingespeist, wenn der Job zum ersten Mal gestartet wird. Wenn das Geheimnis anschließend aktualisiert oder rotiert wird, erhält der Job den aktualisierten Wert nicht automatisch. Sie müssen einen neuen Job starten, um den Service zu zwingen, einen neuen Job mit dem aktualisierten geheimen Wert zu starten.

Erforderliche IAM-Berechtigungen für Geheimnisse AWS Batch

Um diese Funktion verwenden zu können, müssen Sie über die Ausführungsrolle verfügen und in Ihrer Jobdefinition darauf verweisen. Dies ermöglicht dem Container-Agent das Abrufen der erforderlichen Secrets Manager-Ressourcen. Weitere Informationen finden Sie unter AWS Batch IAM-Ausführungsrolle.

Um Zugriff auf die Secrets Manager Manager-Geheimnisse zu gewähren, die Sie erstellen, fügen Sie der Ausführungsrolle manuell die folgenden Berechtigungen als Inline-Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen und Entfernen von IAM-Richtlinien im IAM-Benutzerhandbuch.

- `secretsmanager:GetSecretValue`: Erforderlich, wenn Sie auf ein Secrets Manager-Secret verweisen.
- `kms:Decrypt`: Nur erforderlich, wenn Ihr Secret einen benutzerdefinierten KMS-Schlüssel verwendet und nicht den Standardschlüssel. Der ARN für Ihren benutzerdefinierten Schlüssel sollte als Ressource hinzugefügt werden.

Das folgende Beispiel einer Inline-Richtlinie fügt die erforderlichen Berechtigungen hinzu:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Injizieren sensibler Daten als Umgebungsvariable

In Ihrer Jobdefinition können Sie Folgendes angeben:

- Das `secrets` Objekt, das den Namen der Umgebungsvariablen enthält, die im Job festgelegt werden soll
- Der Amazon-Ressourcenname (ARN) des Secrets Manager-Secrets
- Zusätzliche Parameter, die die sensiblen Daten enthalten, die dem Job präsentiert werden sollen

Das folgende Beispiel zeigt die vollständige Syntax, die für das Secrets Manager-Secret angegeben werden muss.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-
stage:version-id
```

Im folgenden Abschnitt werden die zusätzlichen Parameter beschrieben. Diese Parameter sind optional. Wenn Sie sie jedoch nicht verwenden, müssen Sie die Doppelpunkte angeben, `:` um die Standardwerte zu verwenden. Beispiele finden Sie unten für weiteren Kontext.

`json-key`

Gibt den Namen des Schlüssels in einem Schlüssel-Wert-Paar mit dem Wert an, den Sie als Umgebungsvariablenwert festlegen möchten. Nur Werte im JSON-Format werden unterstützt. Wenn Sie keinen JSON-Schlüssel angeben, wird der vollständige Inhalt des Secrets verwendet.

`version-stage`

Gibt die Phasenbeschriftung der Version eines Secrets an, die Sie verwenden möchten. Wenn eine Versionsphasenbeschriftung angegeben ist, können Sie keine Versions-ID angeben. Wenn keine Versionsphase angegeben wird, besteht das Standardverhalten darin, das Secret Schlüssel mit der `AWSCURRENT`-Phasenbeschriftung abzurufen.

Phasenbeschriftungen werden verwendet, um verschiedene Versionen eines Secrets zu verfolgen, wenn sie aktualisiert oder rotiert werden. Jede Version eines Secrets hat eine oder mehrere Phasenbeschriftungen und eine ID. Weitere Informationen finden Sie unter Wichtige Begriffe und Konzepte für AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

`version-id`

Gibt die eindeutige ID der Version des Secrets an, die Sie verwenden möchten. Wenn eine Versions-ID angegeben wird, können Sie keine Versionsphasenbeschriftung angeben. Wenn keine Versions-ID angegeben wird, besteht das Standardverhalten darin, den geheimen Schlüssel mit der `AWSCURRENT`-Phasenbeschriftung abzurufen.

Versionen IDs werden verwendet, um den Überblick über verschiedene Versionen eines Geheimnisses zu behalten, wenn sie entweder aktualisiert oder ausgetauscht werden. Jede Version eines Secrets hat eine ID. Weitere Informationen finden Sie unter Wichtige Begriffe und Konzepte für AWS Secrets Manager im AWS Secrets Manager Benutzerhandbuch.

Beispiel-Containerdefinitionen

Die folgenden Beispiele zeigen, wie Sie auf Secrets Manager-Secrets in Ihren Containerdefinitionen verweisen können.

Example Verweisen auf ein vollständiges Secret

Im Folgenden finden Sie einen Ausschnitt einer Aufgabendefinition mit dem Format beim Verweisen auf den vollständigen Text eines Secrets Manager-Secret.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
    }]
```

```
  }]
}
```

Example Verweisen auf einen bestimmten Schlüssel innerhalb eines Secrets

Im Folgenden wird ein Beispiel für die Ausgabe eines get-secret-valueBefehls gezeigt, der den Inhalt eines Secrets zusammen mit der zugehörigen Staging-Bezeichnung und der zugehörigen Versions-ID anzeigt.

```
{
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
    "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\",
\"username3\":\"password3\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreatedDate": 1581968848.921
}
```

Verweisen Sie auf einen bestimmten Schlüssel aus der vorherigen Ausgabe in einer Containerdefinition, indem Sie den Schlüsselnamen am Ende des ARN angeben.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::"
    }]
  }]
}
```

Example Verweisen auf eine bestimmte Secret-Version

Im Folgenden wird eine Beispielausgabe eines describe-secret -Befehls gezeigt, der den unverschlüsselten Inhalt eines Secrets zusammen mit den Metadaten für alle Versionen des Secrets anzeigt.

```
{
```

```
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "Description": "Example of a secret containing application authorization data.",
    "RotationEnabled": false,
    "LastChangedDate": 1581968848.926,
    "LastAccessedDate": 1581897600.0,
    "Tags": [],
    "VersionIdsToStages": {
        "871d9eca-18aa-46a9-8785-981dd39ab30c": [
            "AWSCURRENT"
        ],
        "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [
            "AWSPREVIOUS"
        ]
    }
}
```

Verweisen Sie auf eine bestimmte Versionsphasenbeschriftung aus der vorherigen Ausgabe in einer Containerdefinition, indem Sie den Schlüsselnamen am Ende des ARN angeben.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::AWSPREVIOUS:"
    }]
  }]
}
```

Verweisen Sie auf eine bestimmte Versions-ID der vorherigen Ausgabe in einer Containerdefinition, indem Sie den Schlüsselnamen am Ende des ARN angeben.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Example Verweisen auf einen bestimmten Schlüssel und eine Versionsphasenbeschriftung eines Secrets

Im Folgenden wird gezeigt, wie Sie sowohl auf einen bestimmten Schlüssel innerhalb eines Secrets als auch auf eine bestimmte Versionsphasenbeschriftung verweisen.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}
```

Verwenden Sie die folgende Syntax, um einen bestimmten Schlüssel und eine Versions-ID anzugeben.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Fügen Sie vertrauliche Daten in eine Protokollkonfiguration ein

Wenn Sie `logConfiguration` in Ihrer Jobdefinition a angeben, können Sie dies `secretOptions` mit dem Namen der Protokolltreiberoption angeben, die im Container festgelegt werden soll, und dem vollständigen ARN des Secrets Manager-Geheimnisses, das die sensiblen Daten enthält, die dem Container präsentiert werden sollen.

Im Folgenden finden Sie einen Ausschnitt aus einer Jobdefinition, die das Format zeigt, wenn auf ein Secrets Manager Manager-Geheimnis verwiesen wird.

```
{
  "containerProperties": [{
```

```
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
      }]
    }]
  }]
}
```

Erstellen Sie ein Geheimnis AWS Secrets Manager

Sie können die Secrets Manager-Konsole verwenden, um ein Secret für Ihre sensiblen Daten zu erstellen. Weitere Informationen finden Sie unter Erstellen eines Basic Secrets im AWS Secrets Manager -Benutzerhandbuch.

So erstellen Sie ein Basis-Secret

Verwenden Sie Secrets Manager zum Erstellen eines Secrets für Ihre sensiblen Daten.

1. Öffnen Sie die Secrets Manager Manager-Konsole unter https://console.aws.amazon.com/secretsmanager/.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Wählen Sie für Select secret type (Secret-Typ auswählen) die Option Other type of secrets (Anderer Secret-Typ) aus.
4. Geben Sie die Daten Ihres benutzerdefinierten Secrets in Form von Paaren aus Key (Schlüssel) und Value (Wert) an. Sie können z. B. einen Schlüssel von `UserName` festlegen und dann den entsprechenden Benutzernamen als Wert angeben. Fügen Sie einen zweiten Schlüssel mit dem Namen `Password` und dem Passworttext als Wert hinzu. Sie können auch Einträge für einen Datenbanknamen, eine Serveradresse oder einen TCP-Port hinzufügen. Sie können so viele Paare wie nötig zum Speichern der von Ihnen benötigten Informationen hinzufügen.

   Alternativ können Sie die Registerkarte Plaintext (Klartext) wählen und den Secret-Wert auf beliebige Weise eingeben.
5. Wählen Sie den AWS KMS Verschlüsselungsschlüssel aus, den Sie verwenden möchten, um den geschützten Text im Secret zu verschlüsseln. Wenn Sie keinen Schlüssel wählen, prüft

Secrets Manager, ob es einen Standardschlüssel für das Konto gibt. Fall dies der Fall ist, wird er verwendet. Wenn kein Standardschlüssel vorhanden ist, erstellt Secrets Manager automatisch einen Schlüssel. Sie können auch Add new key (Neuen Schlüssel hinzufügen) auswählen, um einen benutzerdefinierten KMS-Schlüssel speziell für dieses Secret zu erstellen. Um Ihren eigenen KMS-Schlüssel zu erstellen, müssen Sie in Ihrem Konto zum Erstellen von KMS-Schlüssel berechtigt sein.

6. Wählen Sie Weiter aus.
7. Geben Sie für Secret name (Secret-Name) einen optionalen Pfad und Namen ein, z. B. **`production/MyAwesomeAppSecret`** oder **`development/TestSecret`**, und wählen Sie Next (Weiter) aus. Optional können Sie eine Beschreibung hinzufügen. Mit dieser können Sie sich später besser an den Zweck dieses Secrets erinnern.

   Der Secret-Name darf nur ASCII-Zeichen, Ziffern oder eines der folgenden Zeichen enthalten: /_+=.@-
8. (Optional) Jetzt können Sie die Rotation für Ihr Secret konfigurieren. Belassen Sie die Option für dieses Verfahren auf Disable automatic rotation (Automatisches Rotieren deaktivieren) und klicken Sie auf Next (Weiter).

   Informationen dazu, wie Sie die Rotation für neue oder bestehende Geheimnisse konfigurieren, finden Sie unter Rotation Ihrer AWS Secrets Manager Geheimnisse.
9. Überprüfen Sie die Einstellungen, und wählen Sie dann Store secret (Secret speichern) aus, um Ihre gesamte Eingabe als neues Secret im Secrets Manager zu speichern.

Geben Sie vertrauliche Daten mit Systems Manager Parameter Store an

Mit können Sie vertrauliche Daten in Ihre Container einfügen AWS Batch, indem Sie Ihre vertraulichen Daten in AWS Systems Manager Parameter Store-Parametern speichern und sie dann in Ihrer Containerdefinition referenzieren.

Themen

- Überlegungen bei der Angabe vertraulicher Daten mithilfe des Systems Manager Manager-Parameterspeichers
- Erforderliche IAM-Berechtigungen für Geheimnisse AWS Batch
- Fügen Sie vertrauliche Daten als Umgebungsvariable ein
- Fügen Sie vertrauliche Daten in eine Protokollkonfiguration ein

- Erstellen Sie einen Parameter Store-Parameter AWS Systems Manager

Überlegungen bei der Angabe vertraulicher Daten mithilfe des Systems Manager Manager-Parameterspeichers

Bei der Angabe sensibler Daten für Container mit Systems Manager-Parameter Store-Parametern sollte Folgendes berücksichtigt werden.

- Für diese Funktion muss auf Ihrer Container-Instance Version 1.23.0 oder höher des Container-Agents installiert sein. Wir empfehlen jedoch, die neueste Container-Agent-Version zu verwenden. Informationen zur Überprüfung Ihrer Agentenversion und zur Aktualisierung auf die neueste Version finden Sie unter Aktualisieren des Amazon ECS-Container-Agenten im Amazon Elastic Container Service Developer Guide.
- Vertrauliche Daten werden in den Container für Ihren Job eingespeist, wenn der Container zum ersten Mal gestartet wird. Wenn der geheime oder der Parameter Store-Parameter aktualisiert oder rotiert wird, erhält der Container nicht automatisch den aktualisierten Wert. Sie müssen einen neuen Job starten, um den Start eines neuen Jobs mit aktualisierten Geheimnissen zu erzwingen.

Erforderliche IAM-Berechtigungen für Geheimnisse AWS Batch

Um diese Funktion verwenden zu können, müssen Sie über die Ausführungsrolle verfügen und in Ihrer Jobdefinition darauf verweisen. Dadurch kann der Amazon ECS-Container-Agent die erforderlichen AWS Systems Manager Ressourcen abrufen. Weitere Informationen finden Sie unter AWS Batch IAM-Ausführungsrolle.

Um Zugriff auf die AWS Systems Manager Parameter Store-Parameter zu gewähren, die Sie erstellen, fügen Sie der Ausführungsrolle manuell die folgenden Berechtigungen als Inline-Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen und Entfernen von IAM-Richtlinien im IAM-Benutzerhandbuch.

- `ssm:GetParameters`: Erforderlich, wenn in einer Aufgabendefinition auf einen Parameter Store-Parameter von Systems Manager verwiesen wird.
- `secretsmanager:GetSecretValue`: Erforderlich, wenn Sie direkt auf ein Secrets Manager-Secret verweisen oder wenn der Parameter Systems Manager Parameterspeicher in einer Aufgabendefinition auf ein Secrets Manager-Secret verweist.

- `kms:Decrypt` – Nur erforderlich, wenn Ihr Geheimnis einen benutzerdefinierten KMS-Schlüssel verwendet und nicht den Standardschlüssel. Der ARN für Ihren benutzerdefinierten Schlüssel sollte als Ressource hinzugefügt werden.

Das folgende Beispiel einer Inline-Richtlinie fügt die erforderlichen Berechtigungen hinzu:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:<region>:<aws_account_id>:parameter/<parameter_name>",
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Fügen Sie vertrauliche Daten als Umgebungsvariable ein

Geben Sie in Ihrer Containerdefinition `secrets` mit dem Namen der im Container zu setzenden Umgebungsvariablen und dem Namen oder dem ARN des Systems Manager-Parameter Store-Parameters an, der die sensiblen Daten enthält, die dem Container präsentiert werden sollen.

Im Folgenden finden Sie einen Ausschnitt aus einer Aufgabendefinition, die das Format beim Verweisen auf einen Systems Manager Manager-Parameter Store-Parameter zeigt. Wenn der Systems Manager Manager-Parameter Store-Parameter in derselben Region wie die Aufgabe vorhanden ist, die Sie starten, können Sie entweder den vollständigen ARN oder den Namen des Parameters verwenden. Wenn der Parameter in einer anderen Region existiert, muss der volle ARN angegeben werden.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Fügen Sie vertrauliche Daten in eine Protokollkonfiguration ein

Beim Angeben von `logConfiguration` können Sie `secretOptions` in Ihrer Containerdefinition mit dem Namen der im Container festzulegenden Protokolltreiberoption und dem vollständigen ARN des Systems Manager-Parameter Store-Parameters angeben, in denen die sensiblen Daten enthalten sind, die dem Container präsentiert werden sollen.

> **Important**
>
> Wenn der Systems Manager Manager-Parameter Store-Parameter in derselben Region wie die Aufgabe vorhanden ist, die Sie starten, können Sie entweder den vollständigen ARN oder den Namen des Parameters verwenden. Wenn der Parameter in einer anderen Region existiert, muss der volle ARN angegeben werden.

Im Folgenden finden Sie einen Ausschnitt aus einer Aufgabendefinition, die das Format beim Verweisen auf einen Systems Manager Manager-Parameter Store-Parameter zeigt.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
      }]
    }]
  }]
```

```
}
```

Erstellen Sie einen Parameter Store-Parameter AWS Systems Manager

Sie können die AWS Systems Manager Konsole verwenden, um einen Systems Manager Parameter Store-Parameter für Ihre vertraulichen Daten zu erstellen. Weitere Informationen finden Sie unter Walkthrough: Erstellen und Verwenden eines Parameter in einem Befehl (Konsole) im AWS Systems Manager -Benutzerhandbuch.

So erstellen Sie einen Parameter Store-Parameter

1. Öffnen Sie die AWS Systems Manager Konsole unter https://console.aws.amazon.com/systems-manager/.
2. Wählen Sie im Navigationsbereich Parameter Store (Parameterspeicher) Create parameter (Parameter erstellen).
3. Geben Sie unter Name eine Hierarchie und einen Parameternamen ein. Geben Sie beispielsweise `test/database_password` ein.
4. Geben Sie unter Description (Beschreibung) eine optionale Beschreibung ein.
5. Wählen Sie als Typ die Option Zeichenfolge StringList, oder aus SecureString.

> **Note**
>
> - Wenn Sie möchten SecureString, wird das Feld KMS-Schlüssel-ID angezeigt. Wenn Sie keine KMS Schlüssel-ID, keinen KMS Schlüssel-ARN, keinen Aliasnamen oder Alias-ARN angeben, verwendet das System `alias/aws/ssm`. Das ist der Standard-KMS-Schlüssel für Systems Manager. Wenn Sie diesen Schlüssel nicht verwenden möchten, können Sie einen benutzerdefinierten Schlüssel auswählen. Weitere Informationen finden Sie unter Verwenden von sicheren String-Parametern im AWS Systems Manager -Benutzerhandbuch.
> - Wenn Sie einen sicheren Stringparameter in der Konsole mit dem Parameter `key-id` entweder mit einem benutzerdefinierten KMS-Schlüssel-Aliasnamen oder einem Alias-ARN erstellen, müssen Sie das Präfix `alias/` vor dem Alias angeben. Nachfolgend ein ARN-Beispiel:
>
> ```
> arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
> ```
>
> Im Folgenden finden Sie ein Beispiel für einen Aliasnamen:

```
alias/MyAliasName
```

6. Geben Sie für Value (Wert) einen Wert ein. Beispiel, `MyFirstParameter`. Wenn Sie sich dafür entscheiden SecureString, wird der Wert genau so maskiert, wie Sie ihn eingegeben haben.
7. Wählen Sie Parameter erstellen aus.

Authentifizierung in der privaten Registrierung für Jobs

AWS Secrets Manager Mithilfe der privaten Registerauthentifizierung für Jobs können Sie Ihre Anmeldeinformationen sicher speichern und sie dann in Ihrer Jobdefinition referenzieren. Auf diese Weise können Sie in Ihren Jobdefinitionen auf Container-Images verweisen, die in privaten Registern existieren AWS , für die keine Authentifizierung erforderlich ist. Diese Funktion wird von Jobs unterstützt, die auf EC2 Amazon-Instances und Fargate gehostet werden.

> **Important**
>
> Wenn Ihre Jobdefinition auf ein Bild verweist, das in Amazon ECR gespeichert ist, trifft dieses Thema nicht zu. Weitere Informationen finden Sie unter Amazon ECR-Images mit Amazon ECS im Amazon Elastic Container-Registry-Benutzerhandbuch.

Für Jobs, die auf EC2 Amazon-Instances gehostet werden, ist für diese Funktion eine Version `1.19.0` oder eine neuere Version des Container-Agenten erforderlich. Wir empfehlen jedoch, die neueste Container-Agent-Version zu verwenden. Informationen dazu, wie Sie Ihre Agentenversion überprüfen und auf die neueste Version aktualisieren können, finden Sie unter Aktualisieren des Amazon ECS-Container-Agenten im Amazon Elastic Container Service Developer Guide.

Für Jobs, die auf Fargate gehostet werden, ist für diese Funktion eine Plattformversion `1.2.0` oder höher erforderlich. Weitere Informationen finden Sie unter Versionen der AWS Fargate Linux-Plattform im Amazon Elastic Container Service Developer Guide.

Geben Sie in Ihrer Containerdefinition das Objekt `repositoryCredentials` mit den Details des von Ihnen erstellten Geheimnisses an. Das Geheimnis, auf das Sie verweisen, kann von einem anderen AWS-Region oder einem anderen Konto als dem Job stammen, der es verwendet.

> **Note**
>
> Wenn Sie die AWS Batch API oder das AWS SDK verwenden und das Geheimnis in demselben AWS-Region Job enthalten ist, den Sie starten, können Sie entweder den vollständigen ARN oder den Namen des Geheimnisses verwenden. AWS CLI Wenn das Geheimnis in einem anderen Konto vorhanden ist, muss der vollständige ARN des Geheimnisses angegeben werden. Bei Verwendung von muss immer der vollständige ARN des Geheimnisses angegeben werden. AWS Management Console

Im Folgenden finden Sie einen Auszug aus einer Jobdefinition, in der die erforderlichen Parameter aufgeführt sind:

```
"containerProperties": [
  {
    "image": "private-repo/private-image",
    "repositoryCredentials": {
      "credentialsParameter":
 "arn:aws:secretsmanager:region:123456789012:secret:secret_name"
    }
  }
]
```

Erforderliche IAM-Berechtigungen für die private Registrierungsauthentifizierung

Die Ausführungsrolle ist erforderlich, um diese Funktion verwenden zu können. Auf diese Weise kann der Container-Agent das Container-Image abrufen. Weitere Informationen finden Sie unter AWS Batch IAM-Ausführungsrolle.

Um Zugriff auf die von Ihnen erstellten Geheimnisse zu gewähren, fügen Sie der Ausführungsrolle die folgenden Berechtigungen als Inline-Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen und Entfernen von IAM-Richtlinien.

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`: Nur erforderlich, wenn Ihr Schlüssel einen benutzerdefinierten KMS-Schlüssel verwendet und nicht den Standard-Schlüssel. Der Amazon-Ressourcenname (ARN) für Ihren benutzerdefinierten Schlüssel muss als Ressource hinzugefügt werden.

Das folgende Beispiel einer Inline-Richtlinie fügt die Berechtigungen hinzu:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name",
        "arn:aws:kms:region:123456789012:key/key_id"
      ]
    }
  ]
}
```

Tutorial: Erstellen Sie ein Geheimnis für die Authentifizierung in privaten Registern

Führen Sie die folgenden Schritte aus, um ein Geheimnis für Ihre privaten Registrierungsdaten mit zu erstellen AWS Secrets Manager.

Erstellen Sie ein grundlegendes Geheimnis

1. Öffnen Sie die AWS Secrets Manager Konsole unter https://console.aws.amazon.com/secretsmanager/.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Wählen Sie für Select secret type (Secret-Typ auswählen) die Option Other type of secrets (Anderer Secret-Typ) aus.
4. Wählen Sie Plaintext (Klartext) aus und geben Sie Ihre privaten Registrierungsanmeldeinformationen in folgendem Format ein:

```
{
```

```
  "username" : "privateRegistryUsername",
  "password" : "privateRegistryPassword"
}
```

5. Wählen Sie Weiter aus.
6. Geben Sie für Secret name (Secret-Name) einen optionalen Pfad und Namen ein, z. B. **`production/MyAwesomeAppSecret`** oder **`development/TestSecret`**, und wählen Sie Next (Weiter) aus. Optional können Sie eine Beschreibung hinzufügen. Mit dieser können Sie sich später besser an den Zweck dieses Secrets erinnern.

   Der Secret-Name darf nur ASCII-Zeichen, Ziffern oder eines der folgenden Zeichen enthalten: `/_+=.@-`.
7. (Optional) Jetzt können Sie die Rotation für Ihr Secret konfigurieren. Belassen Sie die Option für dieses Verfahren auf Disable automatic rotation (Automatisches Rotieren deaktivieren) und klicken Sie auf Next (Weiter).

   Anweisungen zur Konfiguration der Rotation neuer oder vorhandener Secrets finden Sie unter Rotating Your AWS Secrets Manager Secrets.
8. Überprüfen Sie die Einstellungen, und wählen Sie dann Store secret (Secret speichern) aus, um Ihre gesamte Eingabe als neues Secret im Secrets Manager zu speichern.

Registrieren Sie eine Jobdefinition und aktivieren Sie unter Private Registrierung die Authentifizierung mit privater Registrierung. Geben Sie dann unter Secrets-Manager-ARN oder -Name den Amazon-Ressourcennamen (ARN) des Secrets ein. Weitere Informationen finden Sie unter Erforderliche IAM-Berechtigungen für die private Registrierungsauthentifizierung.

Amazon EFS-Volumes

Amazon Elastic File System (Amazon EFS) bietet einfachen, skalierbaren Dateispeicher für Ihre AWS Batch Jobs. Mit Amazon EFS ist die Speicherkapazität elastisch. Es wird automatisch skaliert, wenn Sie Dateien hinzufügen und entfernen. Ihre Anwendungen verfügen jederzeit über den jeweils erforderlichen Speicherplatz.

Sie können Amazon EFS-Dateisysteme mit verwenden AWS Batch , um Dateisystemdaten aus Ihrer gesamten Flotte von Container-Instances zu exportieren. Auf diese Weise haben Ihre Jobs Zugriff auf denselben persistenten Speicher. Sie müssen Ihr Container-Instance-AMI jedoch konfigurieren, um das Amazon EFS-Dateisystem zu mounten, bevor der Docker-Daemon startet. Außerdem müssen

Ihre Jobdefinitionen auf Volume-Mounts auf der Container-Instance verweisen, um das Dateisystem verwenden zu können. Die folgenden Abschnitte helfen Ihnen bei den ersten Schritten mit Amazon EFS AWS Batch.

Überlegungen zu Amazon EFS Volumes

Bei der Verwendung von Amazon EFS-Volumes sollte Folgendes berücksichtigt werden:

- Für Jobs, die EC2 Ressourcen verwenden, wurde die Unterstützung des Amazon EFS-Dateisystems als öffentliche Vorschauversion mit der für Amazon ECS optimierten AMI-Version `20191212` mit der Container-Agent-Version 1.35.0 hinzugefügt. Die Amazon EFS-Dateisystemunterstützung wurde jedoch mit der für Amazon ECS optimierten AMI-Version `20200319` mit der Container-Agent-Version 1.38.0, die den Amazon EFS Access Point und die IAM-Autorisierungsfunktionen enthielt, allgemein verfügbar. Wir empfehlen, dass Sie die für Amazon ECS optimierte AMI-Version `20200319` oder höher verwenden, um diese Funktionen nutzen zu können. Weitere Informationen finden Sie unter Amazon ECS-optimierte AMI-Versionen im Amazon Elastic Container Service Developer Guide.

  > **Note**
  >
  > Wenn Sie Ihr eigenes AMI erstellen, müssen Sie den Container-Agenten 1.38.0 oder höher, `ecs-init` Version 1.38.0-1 oder höher verwenden und die folgenden Befehle auf Ihrer Amazon-Instance ausführen. EC2 Dies ist alles, um das Amazon ECS-Volumen-Plugin zu aktivieren. Die Befehle hängen davon ab, ob Sie Amazon Linux 2 oder Amazon Linux als Basis-Image verwenden.
  >
  > Amazon Linux 2
  >
  > ```
  > $ yum install amazon-efs-utils
  > systemctl enable --now amazon-ecs-volume-plugin
  > ```
  >
  > Amazon Linux
  >
  > ```
  > $ yum install amazon-efs-utils
  > sudo shutdown -r now
  > ```

- Für Jobs, die Fargate-Ressourcen verwenden, wurde Amazon EFS-Dateisystemunterstützung hinzugefügt, wenn Plattformversion 1.4.0 oder höher verwendet wird. Weitere Informationen finden Sie unter AWS Fargate-Plattformversionen im Amazon Elastic Container Service Developer Guide.

- Bei der Angabe von Amazon EFS-Volumes in Aufträgen, die Fargate-Ressourcen verwenden, erstellt Fargate einen Supervisor-Container, der für die Verwaltung des Amazon EFS-Volumes verantwortlich ist. Der Supervisor-Container verwendet einen kleinen Teil des Arbeitsspeichers. Der Supervisor-Container ist beim Abfragen des Endpunkts der Aufgabenmetadaten Version 4 sichtbar. Weitere Informationen finden Sie unter Aufgabenmetadaten-Endpunkt Version 4 im Amazon Elastic Container Service-Benutzerhandbuch für AWS Fargate.

Verwenden Sie Amazon EFS-Zugriffspunkte

Amazon EFS Access Points sind anwendungsspezifische Einstiegspunkte in ein EFS-Dateisystem, mit denen Sie den Anwendungszugriff auf gemeinsam genutzte Datensätze verwalten können. Weitere Informationen zu Amazon EFS Access Points und zur Steuerung des Zugriffs auf diese finden Sie unter Arbeiten mit Amazon EFS Access Points im Amazon Elastic File System-Benutzerhandbuch.

Zugriffspunkte können eine Benutzeridentität, einschließlich der POSIX-Gruppen des Benutzers, für alle Dateisystemanforderungen erzwingen, die über den Zugriffspunkt erfolgen. Zugriffspunkte können auch ein anderes Stammverzeichnis für das Dateisystem erzwingen, so dass Clients nur auf Daten im angegebenen Verzeichnis oder in seinen Unterverzeichnissen zugreifen können.

> **Note**
>
> Beim Erstellen eines EFS-Zugriffspunkts geben Sie einen Pfad im Dateisystem an, der als Stammverzeichnis dienen soll. Wenn Sie in Ihrer AWS Batch Jobdefinition mit einer Zugriffspunkt-ID auf das EFS-Dateisystem verweisen, muss das Stammverzeichnis entweder weggelassen oder auf / Dies erzwingt den Pfad gesetzt werden, der auf dem EFS-Zugriffspunkt festgelegt ist.

Sie können eine AWS Batch IAM-Job-Rolle verwenden, um zu erzwingen, dass bestimmte Anwendungen einen bestimmten Zugriffspunkt verwenden. Durch die Kombination von IAM-Richtlinien mit Zugriffspunkten können Sie ganz einfach einen sicheren Zugriff auf bestimmte Datasets für Ihre Anwendungen bereitstellen. Diese Funktion verwendet Amazon ECS IAM-Rollen für Aufgabenfunktionen. Weitere Informationen finden Sie unter IAM-Rollen für Aufgaben im Entwicklerhandbuch zum Amazon Elastic Container Service.

Geben Sie in Ihrer Jobdefinition ein Amazon EFS-Dateisystem an

Um Amazon EFS-Dateisystem-Volumes für Ihre Container zu verwenden, müssen Sie die Volume- und Bereitstellungspunktkonfigurationen in Ihrer Auftragsdefinition angeben. Das folgende JSON-Snippet mit der Jobdefinition zeigt die Syntax für die `mountPoints` Objekte `volumes` und für einen Container:

```
{
    "containerProperties": [
        {
            "image": "amazonlinux:2",
            "command": [
                "ls",
                "-la",
                "/mount/efs"
            ],
            "mountPoints": [
                {
                    "sourceVolume": "myEfsVolume",
                    "containerPath": "/mount/efs",
                    "readOnly": true
                }
            ],
            "volumes": [
                {
                    "name": "myEfsVolume",
                    "efsVolumeConfiguration": {
                        "fileSystemId": "fs-12345678",
                        "rootDirectory": "/path/to/my/data",
                        "transitEncryption": "ENABLED",
                        "transitEncryptionPort": integer,
                        "authorizationConfig": {
                            "accessPointId": "fsap-1234567890abcdef1",
                            "iam": "ENABLED"
                        }
                    }
                }
            ]
        }
    ]
}
```

`efsVolumeConfiguration`

Typ: Objekt

Erforderlich: Nein

Dieser Parameter wird nur bei der Verwendung von Amazon EFS-Volumes angegeben.

`fileSystemId`

Typ: Zeichenfolge

Erforderlich: Ja

Die zu verwendende Amazon EFS-Dateisystem-ID.

`rootDirectory`

Typ: Zeichenfolge

Erforderlich: Nein

Das Verzeichnis im Amazon EFS-Dateisystem, das als Stammverzeichnis im Host bereitgestellt werden soll. Wenn dieser Parameter weggelassen wird, wird der Stamm des Amazon-EFS-Volumes verwendet. Die Angabe von `/` hat denselben Effekt wie das Weglassen dieses Parameters. Er kann bis zu 4.096 Zeichen lang sein.

> **Important**
>
> Wenn in der ein EFS-Zugriffspunkt angegeben ist`authorizationConfig`, muss der Stammverzeichnisparameter entweder weggelassen oder auf gesetzt werden`/`. Dadurch wird der Pfad durchgesetzt, der auf dem EFS-Zugriffspunkt festgelegt ist.

`transitEncryption`

Typ: Zeichenfolge

Zulässige Werte: `ENABLED` | `DISABLED`

Erforderlich: Nein

Legt fest, ob die Verschlüsselung für Amazon EFS-Daten aktiviert werden soll, die zwischen dem AWS Batch Host und dem Amazon EFS-Server übertragen werden. Die Transit-

Verschlüsselung muss aktiviert sein, wenn die Amazon-EFS-IAM-Autorisierung verwendet wird. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert `DISABLED` verwendet. Weitere Informationen finden Sie unter Verschlüsseln von Daten während der Übertragung im Benutzerhandbuch für Amazon Elastic File System.

`transitEncryptionPort`

Typ: Ganzzahl

Erforderlich: Nein

Der Port, der beim Senden verschlüsselter Daten zwischen dem AWS Batch Host und dem Amazon EFS-Server verwendet werden soll. Wenn Sie keinen Transit-Verschlüsselungsport angeben, wird die Port-Auswahlstrategie verwendet, die der Amazon EFS-Mount-Helfer verwendet. Dieser Wert muss zwischen 0 und 65.535 liegen. Weitere Informationen finden Sie unter EFS-Mount-Helfer im Benutzerhandbuch für Amazon Elastic File System.

`authorizationConfig`

Typ: Objekt

Erforderlich: Nein

Die Autorisierungskonfigurationsdetails für das Amazon EFS-Dateisystem.

`accessPointId`

Typ: Zeichenfolge

Erforderlich: Nein

Die zu verwendende Zugriffspunkt-ID. Wenn ein Zugriffspunkt angegeben ist, `efsVolumeConfiguration` muss der Stammverzeichniswert in der entweder weggelassen oder auf gesetzt werden`/`. Dadurch wird der Pfad durchgesetzt, der auf dem EFS-Zugriffspunkt festgelegt ist. Wenn ein Zugriffspunkt verwendet wird, muss in `EFSVolumeConfiguration` die Transitverschlüsselung aktiviert sein. Weitere Informationen finden Sie unter Arbeiten mit Amazon EFS-Zugriffspunkten im Amazon Elastic File System-Benutzerhandbuch.

`iam`

Typ: Zeichenfolge

Zulässige Werte: `ENABLED` | `DISABLED`

Erforderlich: Nein

Legt fest, ob beim Mounten des Amazon EFS-Dateisystems die in einer Auftragsdefinition definierte AWS Batch Job-IAM-Rolle verwendet werden soll. Wenn diese Option aktiviert ist, muss die Transit-Verschlüsselung in `EFSVolumeConfiguration` aktiviert sein. Wenn dieser Parameter nicht angegeben ist, wird der Standardwert `DISABLED` verwendet. Weitere Informationen zur IAM-Ausführungsrolle finden Sie unter. AWS Batch IAM-Ausführungsrolle

Referenz: Beispiele für Berufsdefinitionen

Die Beispiele für Auftragsdefinitionen in den folgenden Themen veranschaulichen die Verwendung gängiger Muster wie Umgebungsvariablen, Parameterersetzung und Volumenzuweisung.

Inhalt

- Referenz: Umgebungsvariablen
- Referenz: Parametersetzung
- Referenz: Testen Sie die GPU-Funktionalität
- Referenz: Paralleljob mit mehreren Knoten

Referenz: Umgebungsvariablen

Die folgende Beispiel-Auftragsdefinition verwendet Umgebungsvariablen, um einen Dateityp und eine Amazon S3 S3-URL anzugeben. Dieses spezielle Beispiel stammt aus dem Blogbeitrag Creating a Simple „Fetch & Run“ AWS Batch Job Compute. Das im Blogbeitrag beschriebene `fetch_and_run.sh`Skript verwendet diese Umgebungsvariablen, um das `myjob.sh` Skript von S3 herunterzuladen und seinen Dateityp zu deklarieren.

Obwohl die Befehls- und Umgebungsvariablen in diesem Beispiel fest in der Jobdefinition kodiert sind, können Sie Überschreibungen für Befehle und Umgebungsvariablen angeben, um die Jobdefinition vielseitiger zu gestalten.

```
{
    "jobDefinitionName": "fetch_and_run",
    "type": "container",
    "containerProperties": {
```

```
        "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "2000"
            },
            {
                "type": "VCPU",
                "value": "2"
            }
        ],
        "command": [
            "myjob.sh",
            "60"
        ],
        "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
        "environment": [
            {
                "name": "BATCH_FILE_S3_URL",
                "value": "s3://amzn-s3-demo-source-bucket/myjob.sh"
            },
            {
                "name": "BATCH_FILE_TYPE",
                "value": "script"
            }
        ],
        "user": "nobody"
    }
}
```

Referenz: Parametersetzung

Im folgenden Auftragsdefinitionsbeispiel wird gezeigt, wie Sie die Parameterersetzung zulassen und Standardwerte festlegen.

Die `Ref::`-Deklarationen im Abschnitt `command` werden zum Festlegen von Platzhaltern für die Parameterersetzung verwendet. Wenn Sie einen Auftrag mit dieser Auftragsdefinition senden, geben Sie die Parameterüberschreibungen für diese Werte an, z. B. `inputfile` und `outputfile`. Im folgenden `parameters` Abschnitt wird eine Standardeinstellung für festgelegt`codec`, Sie können diesen Parameter jedoch nach Bedarf überschreiben.

Weitere Informationen finden Sie unter Parameter.

```
{
    "jobDefinitionName": "ffmpeg_parameters",
    "type": "container",
    "parameters": {"codec": "mp4"},
    "containerProperties": {
        "image": "my_repo/ffmpeg",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "2000"
            },
            {
                "type": "VCPU",
                "value": "2"
            }
        ],
        "command": [
            "ffmpeg",
            "-i",
            "Ref::inputfile",
            "-c",
            "Ref::codec",
            "-o",
            "Ref::outputfile"
        ],
        "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
        "user": "nobody"
    }
}
```

Referenz: Testen Sie die GPU-Funktionalität

Mit dem folgende Auftragsdefinitionsbeispiel wird getestet, ob die GPU-Workload-AMI in Verwenden Sie ein GPU-Workload-AMI ordnungsgemäß konfiguriert ist. In dieser Beispiel-Jobdefinition wird das TensorFlow Deep MNIST-Classifier-Beispiel von ausgeführt. GitHub

```
{
    "containerProperties": {
        "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
        "resourceRequirements": [
            {
                "type": "MEMORY",
```

```
                "value": "32000"
            },
            {
                "type": "VCPU",
                "value": "8"
            }
        ],
        "command": [
            "sh",
            "-c",
            "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
        ]
    },
    "type": "container",
    "jobDefinitionName": "tensorflow_mnist_deep"
}
```

Sie können eine Datei mit dem obigen JSON-Text namens erstellen `tensorflow_mnist_deep.json` und dann eine AWS Batch Jobdefinition mit dem folgenden Befehl registrieren:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Referenz: Paralleljob mit mehreren Knoten

Die folgende beispielhafte Aufgabendefinition veranschaulicht eine parallele Aufgabe mit mehreren Knoten. Weitere Informationen finden Sie im AWS Compute-Blog unter Aufbau eines eng gekoppelten Molekulardynamik-Workflows mit AWS Batch parallel Jobs mit mehreren Knoten.

```
{
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-
jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
```

```
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
              {
                  "type": "MEMORY",
                  "value": "24000"
              },
              {
                  "type": "VCPU",
                  "value": "8"
              }
          ],
          "command": [],
          "jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
          "ulimits": [],
          "instanceType": "p3.2xlarge"
        }
      }
    ]
  }
}
```

Aufträge

Jobs sind die Arbeitseinheit, mit der begonnen wurde AWS Batch. Jobs können als containerisierte Anwendungen aufgerufen werden, die auf Amazon ECS-Container-Instances in einem ECS-Cluster ausgeführt werden.

Aufträge in Containern können auf ein Container-Image, einen Befehl und Parameter verweisen. Weitere Informationen finden Sie unter JobDefinition.

Sie können eine große Anzahl von unabhängigen, einfachen Aufträgen senden.

Themen

- Tutorial: Einen Job einreichen
- Auftragsstatus
- AWS Batch Variablen der Arbeitsumgebung
- Automatisierte Auftragswiederholungen
- Abhängigkeiten von Job
- Timeouts bei der Job
- Stellenangebote bei Amazon EKS
- parallel Jobs mit mehreren Knoten
- parallel Jobs mit mehreren Knoten auf Amazon EKS
- Ordnen Sie Jobs an
- Führen Sie GPU-Jobs aus
- Suchen Sie nach AWS Batch Aufträgen in einer Jobwarteschlange
- Netzwerkmodi für AWS Batch Jobs
- Tutorial: AWS Batch Jobprotokolle an CloudWatch Logs senden
- Tutorial: AWS Batch Jobinformationen überprüfen

Tutorial: Einen Job einreichen

Nachdem Sie eine Jobdefinition registriert haben, können Sie sie als Job an AWS Batch eine Job-Warteschlange senden. Sie können viele der Parameter, die in der Auftragsdefinition angegeben sind, zur Laufzeit überschreiben.

So senden Sie einen Auftrag

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich die Option Jobs (Aufträge) aus.
4. Wählen Sie Neuen Job einreichen aus.
5. Geben Sie unter Name einen eindeutigen Namen für Ihre Jobdefinition ein. Der Name kann bis zu 128 Zeichen lang sein. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.
6. Wählen Sie unter Jobdefinition eine bestehende Jobdefinition für Ihren Job aus. Weitere Informationen finden Sie unter Erstellen Sie eine Auftragsdefinition mit einem Knoten .
7. Wählen Sie für Job-Warteschlange eine bestehende Job-Warteschlange aus. Weitere Informationen finden Sie unter Eine Job-Warteschlange erstellen.
8. Wählen Sie für Jobabhängigkeiten die Option Jobabhängigkeiten hinzufügen aus.
   - Geben Sie unter Job-ID die Job-ID für alle Abhängigkeiten ein. Wählen Sie dann Jobabhängigkeiten hinzufügen aus. Ein Job kann bis zu 20 Abhängigkeiten haben. Weitere Informationen finden Sie unter Abhängigkeiten von Job.
9. (Nur Array-Aufträge) Geben Sie für Array-Größe eine Array-Größe zwischen 2 und 10.000 an.
10. (Optional) Erweitern Sie Tags und wählen Sie dann Tag hinzufügen aus, um der Ressource Tags hinzuzufügen. Geben Sie einen Schlüssel und einen optionalen Wert ein und wählen Sie dann Tag hinzufügen.
11. Wählen Sie „Nächste Seite“.
12. Gehen Sie im Abschnitt Job-Overrides wie folgt vor:
    a. (Optional) Geben Sie für Planungspriorität einen Wert zwischen 0 und 100 ein. Höhere Werte erhalten eine höhere Priorität.
    b. (Optional) Geben Sie für Auftragsversuche ein, wie oft maximal AWS Batch versucht wird, den Job in einen `RUNNABLE` Status zu versetzen. Sie können eine Zahl zwischen 1 und 10 eingeben. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.
    c. (Optional) Geben Sie für das Ausführungs-Timeout den Timeout-Wert (in Sekunden) ein. Das Ausführungs-Timeout ist die Zeitspanne, bis ein unvollendeter Job beendet wird. Wenn ein Versuch die Timeout-Dauer überschreitet, wird er gestoppt und erhält einen Status.

`FAILED` Weitere Informationen finden Sie unter Timeouts bei der Job. Der Mindestwert beträgt 60 Sekunden.

> **Important**
>
> Verlassen Sie sich nicht darauf, dass Jobs, die auf Fargate-Ressourcen ausgeführt werden, länger als 14 Tage laufen. Nach 14 Tagen sind die Fargate-Ressourcen möglicherweise nicht mehr verfügbar, und der Job wird wahrscheinlich beendet.

d. (Optional) Aktivieren Sie die Option Tags weitergeben, um Tags aus dem Auftrag und der Auftragsdefinition an die Amazon ECS-Aufgabe weiterzugeben.

13. Erweitern Sie Additional configuration (Zusätzliche Konfiguration).
14. (Optional) Wählen Sie unter Bedingungen für die Wiederholungsstrategie die Option Beim Beenden bewerten hinzufügen aus. Geben Sie mindestens einen Parameterwert ein und wählen Sie dann eine Aktion aus. Für jeden Satz von Bedingungen muss Aktion entweder auf Erneut versuchen oder Beenden gesetzt sein. Diese Aktionen bedeuten Folgendes:

    - Wiederholen — AWS Batch Wiederholt die Versuche, bis die von Ihnen angegebene Anzahl von Auftragsversuchen erreicht ist.
    - Beenden — AWS Batch beendet die Wiederholung des Jobs.

    > **Important**
    >
    > Wenn Sie „Beim Beenden bewerten hinzufügen“ wählen, konfigurieren Sie mindestens einen Parameter und wählen Sie entweder eine Aktion oder „Beim Beenden auswerten entfernen“.

15. Wählen Sie unter Parameter die Option Parameter hinzufügen aus, um Platzhalter für die Parametersetzung hinzuzufügen. Geben Sie dann einen Schlüssel und einen optionalen Wert ein.
16. Gehen Sie im Abschnitt Container-Überschreibungen wie folgt vor:

    a. Geben Sie für Command die Befehle als entsprechende JSON-String-Array-Entsprechung in das Feld ein.

Dieser Parameter ist `Cmd` im Abschnitt Erstellen eines Containers der Docker Remote-API und dem Parameter `COMMAND` von docker run zugeordnet. Weitere Informationen zum `CMD` Docker-Parameter finden Sie unter https://docs.docker.com/engine/reference/builder/ #cmd.

> **Note**
>
> Dieser Parameter darf keine leere Zeichenfolge enthalten.

b. Geben Sie für v CPUs die Zahl von v einCPUs , die für den Container reserviert werden sollen. Dieser Parameter ordnet zu `CpuShares` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--cpu-shares` für die docker run zu. Jede vCPU entspricht 1.024 CPU-Anteilen. Sie müssen mindestens eine vCPU angeben.

c. Geben Sie für Speicher das Speicherlimit ein, das für den Container verfügbar ist. Wenn Ihr Container versucht, den hier angegebenen Speicher zu überschreiten, wird der Container gestoppt. Dieser Parameter ordnet zu `Memory` im Bereich Erstellen eines Containers der Docker Remote API und der Option `--memory` für die docker run zu. Sie müssen mindestens 4 MB Arbeitsspeicher für einen Auftrag festlegen.

> **Note**
>
> Um Ihre Ressourcennutzung zu maximieren, priorisieren Sie den Arbeitsspeicher für Jobs eines bestimmten Instance-Typs. Weitere Informationen finden Sie unter Speicherverwaltung für Rechenressourcen.

d. (Optional) Wählen Sie unter Anzahl von die Anzahl der aus GPUs, die für den Container reserviert werden GPUs soll.

e. (Optional) Wählen Sie für Umgebungsvariablen die Option Umgebungsvariable hinzufügen aus, um Umgebungsvariablen als Name-Wert-Paare hinzuzufügen. Diese Variablen werden an den Container übergeben.

f. Wählen Sie Nächste Seite.

g. Überprüfen Sie für Job review die Konfigurationsschritte. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Jobdefinition erstellen.

Auftragsstatus

Wenn Sie einen AWS Batch Job an eine Auftragswarteschlange weiterleiten, wechselt der Job in den `SUBMITTED` Status. Danach durchläuft er die folgenden Zustände, bis er erfolgreich ist (Abschluss mit Code `0`) oder fehlschlägt (Abschluss mit einem Code ungleich null). AWS Batch -Aufträge können die folgenden Status haben:

`SUBMITTED`

Ein Job, der an die Warteschlange gesendet wurde und noch nicht vom Scheduler bewertet wurde. Der Scheduler überprüft den Auftrag, um zu bestimmen, ob offene Abhängigkeiten von der erfolgreichen Ausführung anderer Aufträge vorliegen. Wenn Abhängigkeiten vorhanden sind, wird der Auftrag in den Status `PENDING` verschoben. Sind keine Abhängigkeiten vorhanden, wechselt der Auftrag in den Status `RUNNABLE`.

`PENDING`

Ein Job, der sich in der Warteschlange befindet und aufgrund einer Abhängigkeit von einem anderen Job oder einer anderen Ressource noch nicht ausgeführt werden kann. Nachdem die Abhängigkeiten erfüllt sind, wird der Auftrag in den Status `RUNNABLE` verschoben.

`RUNNABLE`

Ein Auftrag, der sich in der Warteschlange befindet, keine ausstehenden Abhängigkeiten hat und daher für einen Host geplant werden kann. Jobs in diesem Status werden gestartet, sobald in einer der Rechenumgebungen, die der Warteschlange des Jobs zugeordnet sind, genügend Ressourcen verfügbar sind. Aufträge können jedoch für einen unbegrenzten Zeitraum in diesem Status verbleiben, wenn keine ausreichenden Ressourcen zur Verfügung stehen.

> **Note**
>
> Wenn Ihre Jobs nicht weitergeführt werden`STARTING`, finden Sie Jobs, die in einem `RUNNABLE` Status feststecken im Abschnitt zur Fehlerbehebung weitere Informationen.

`STARTING`

Diese Aufträge wurden für einen Host geplant und die relevanten Vorgänge zur Container-Initiierung werden noch ausgeführt. Nachdem das Container-Image durch Pull-Übertragung abgerufen und der Container gestartet wurde, geht der Auftrag in den Status `RUNNING` über.

Die Dauer des Abrufs von Images, die Dauer der Fertigstellung von Amazon EKS InitContainer und die Dauer der Auflösung von Amazon ECS ContainerDependency erfolgen im Status STARTING. Die Zeit, die zum Abrufen eines Images für Ihren Job benötigt wird, entspricht der Zeit, in der sich Ihr Job im Status STARTING befindet.

Wenn es beispielsweise drei Minuten dauert, das Bild für Ihren Job abzurufen, befindet sich Ihr Job drei Minuten lang im Status START. Wenn die Ausführung von InitContainers insgesamt zehn Minuten dauert, wird Ihr Amazon EKS-Job zehn Minuten lang in STARTING sein. Wenn Sie Amazon ECS-ContainerDependencies-Sets in Ihrem Amazon ECS-Job haben, befindet sich der Job in STARTING, bis alle Container-Abhängigkeiten (ihre Laufzeit) aufgelöst sind. STARTING ist in den Timeouts nicht enthalten; die Dauer beginnt bei RUNNING. Weitere Informationen finden Sie unter Jobstatus.

`RUNNING`

Der Job wird als Container-Job auf einer Amazon ECS-Container-Instance in einer Rechenumgebung ausgeführt. Wenn der Container des Auftrags beendet wird, bestimmt der Prozessbeendigungscode, ob der Auftrag erfolgreich war oder nicht. Der Beendigungscode `0` bedeutet eine erfolgreiche Ausführung und ein Beendigungscode ungleich Null bedeutet, dass ein Fehler aufgetreten ist. Wenn für den mit einem fehlgeschlagenen Versuch verbundene Aufträge verbleibende Versuche in der optionalen Wiederholungsstrategiekonfiguration vorhanden sind, wird der Auftrag erneut in den Status `RUNNABLE` verschoben. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.

> **Note**
>
> Protokolle für `RUNNING` Jobs sind unter CloudWatch Logs verfügbar. Die Protokollgruppe ist`/aws/batch/job`, und das Format des Protokollstream-Namens lautet wie folgt:*`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`*. Dieses Format könnte sich in future ändern.
> Nachdem ein Job den `RUNNING` Status erreicht hat, können Sie seinen Log-Stream-Namen mithilfe der DescribeJobsAPI-Operation programmgesteuert abrufen. Weitere Informationen finden Sie unter An CloudWatch Logs gesendete Protokolldaten anzeigen im Amazon CloudWatch Logs-Benutzerhandbuch. Standardmäßig laufen diese Protokolle nie ab. Sie können den Aufbewahrungszeitraum jedoch ändern. Weitere Informationen finden Sie unter Aufbewahrung von Protokolldaten in CloudWatch Protokollen ändern im Amazon CloudWatch Logs-Benutzerhandbuch.

`SUCCEEDED`

Der Auftrag wurde mit dem Beendigungscode `0` erfolgreich abgeschlossen. Der Auftragsstatus für `SUCCEEDED` Jobs wird AWS Batch für mindestens 7 Tage beibehalten.

> **Note**
>
> Protokolle für `SUCCEEDED` Jobs sind unter CloudWatch Logs verfügbar. Die Protokollgruppe ist `/aws/batch/job`, und das Format des Protokollstream-Namens lautet wie folgt:*`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`*. Dieses Format kann sich in future ändern.
> Nachdem ein Job den `RUNNING` Status erreicht hat, können Sie seinen Log-Stream-Namen mithilfe der DescribeJobsAPI-Operation programmgesteuert abrufen. Weitere Informationen finden Sie unter An CloudWatch Logs gesendete Protokolldaten anzeigen im Amazon CloudWatch Logs-Benutzerhandbuch. Standardmäßig laufen diese Protokolle nie ab. Sie können den Aufbewahrungszeitraum jedoch ändern. Weitere Informationen finden Sie unter Aufbewahrung von Protokolldaten in CloudWatch Protokollen ändern im Amazon CloudWatch Logs-Benutzerhandbuch.

`FAILED`

Der Auftrag wurde bei allen verfügbaren Versuchen nicht erfolgreich beendet. Der Auftragsstatus für `FAILED` Jobs wird AWS Batch für mindestens 7 Tage beibehalten.

> **Note**
>
> Protokolle für `FAILED` Jobs sind unter CloudWatch Logs verfügbar. Die Protokollgruppe ist `/aws/batch/job`, und das Format des Protokollstream-Namens lautet wie folgt:*`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`*. Dieses Format kann sich in future ändern.
> Nachdem ein Job den `RUNNING` Status erreicht hat, können Sie seinen Protokollstream mithilfe der DescribeJobsAPI-Operation programmgesteuert abrufen. Weitere Informationen finden Sie unter An CloudWatch Logs gesendete Protokolldaten anzeigen im Amazon CloudWatch Logs-Benutzerhandbuch. Standardmäßig laufen diese Protokolle nie ab. Sie können den Aufbewahrungszeitraum jedoch ändern. Weitere Informationen finden Sie unter Aufbewahrung von Protokolldaten in CloudWatch Protokollen ändern im Amazon CloudWatch Logs-Benutzerhandbuch.

AWS Batch Variablen der Arbeitsumgebung

AWS Batch legt spezifische Umgebungsvariablen in Container-Jobs fest. Diese Umgebungsvariablen ermöglichen eine Introspektion der Container innerhalb von Jobs. Sie können die Werte dieser Variablen in der Logik Ihrer Anwendungen verwenden. Alle Variablen, die AWS Batch gesetzt werden, beginnen mit dem `AWS_BATCH_` Präfix. Dies ist ein geschütztes Umgebungsvariablenpräfix. Sie können dieses Präfix nicht für Ihre eigenen Variablen in Jobdefinitionen oder Überschreibungen verwenden.

Die folgenden Umgebungsvariablen stehen in Auftragscontainern zur Verfügung:

`AWS_BATCH_CE_NAME`

Diese Variable ist auf den Namen der Rechenumgebung gesetzt, in der Ihr Job platziert ist.

`AWS_BATCH_JOB_ARRAY_INDEX`

Diese Variable wird nur in untergeordneten Array-Aufträgen festgelegt. Der Array-Auftrags-Index beginnt bei 0 und alle untergeordneten Aufträge erhalten eine eindeutige Indexnummer. Beispielsweise weist ein Array-Auftrag mit 10 untergeordneten Aufträge die Indexwerte 0–9 auf. Sie können diesen Indexwert dazu verwenden, zu steuern, wie Ihre untergeordneten Array-Auftragselemente differenziert werden. Weitere Informationen finden Sie unter Tutorial: Verwenden Sie den Array-Jobindex, um die Jobdifferenzierung zu steuern.

`AWS_BATCH_JOB_ARRAY_SIZE`

Diese Variable ist auf die Größe des übergeordneten Array-Jobs gesetzt. Die Größe des übergeordneten Array-Jobs wird in dieser Variablen an den untergeordneten Array-Job übergeben.

`AWS_BATCH_JOB_ATTEMPT`

Für diese Variable wird die Auftragsversuchsnummer festgelegt. Der erste Versuch erhält die Nummer 1. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.

`AWS_BATCH_JOB_ID`

Diese Variable ist auf die AWS Batch Job-ID gesetzt.

`AWS_BATCH_JOB_KUBERNETES_NODE_UID`

Diese Variable wird als Kubernetes UID des Knotenobjekts festgelegt, das sich im Kubernetes-Cluster befindet, auf dem der Pod ausgeführt wird. Diese Variable ist nur für Jobs festgelegt, die

auf Amazon EKS-Ressourcen ausgeführt werden. Weitere Informationen finden Sie UIDsin der KubernetesDokumentation.

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Diese Variable wird nur in parallelen Aufträgen mit mehreren Knoten festgelegt. Diese Variable ist auf die Indexnummer des Hauptknotens des Auftrags festgelegt. Ihr Anwendungscode kann den mit dem `AWS_BATCH_JOB_MAIN_NODE_INDEX` `AWS_BATCH_JOB_NODE_INDEX` auf einem einzelnen Knoten vergleichen, um festzustellen, ob es sich um den Hauptknoten handelt.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Diese Variable wird nur in untergeordneten Knoten für parallel Jobs mit mehreren Knoten gesetzt. Diese Variable ist auf dem Hauptknoten nicht vorhanden, sondern auf die private IPv4 Adresse des Hauptknotens des Jobs gesetzt. Der Anwendungscode Ihres untergeordneten Knotens kann diese Adresse zur Kommunikation mit dem Hauptknoten verwenden.

`AWS_BATCH_JOB_NODE_INDEX`

Diese Variable wird nur in parallelen Aufträgen mit mehreren Knoten festgelegt. Diese Variable ist auf die Indexnummer des Knotens festgelegt. Der Knotenindex beginnt bei 0 und jeder Knoten erhält eine eindeutige Indexnummer. Beispielsweise weist ein paralleler Auftrag mit mehreren Knoten und 10 untergeordneten Aufträgen die Indexwerte 0–9 auf.

`AWS_BATCH_JOB_NUM_NODES`

Diese Variable wird nur in parallelen Aufträgen mit mehreren Knoten festgelegt. Diese Variable ist auf die Anzahl der Knoten gesetzt, die Sie für Ihren Paralleljob mit mehreren Knoten angefordert haben.

`AWS_BATCH_JQ_NAME`

Für diese Variable ist der Name der Auftragswarteschlange festgelegt, an die der Auftrag übermittelt wurde.

Automatisierte Auftragswiederholungen

Sie können auf Ihre Aufträge und Auftragsdefinitionen eine Wiederholungsstrategie anwenden, mit der fehlgeschlagenen Aufträge automatisch wiederholt werden. Zu den möglichen Ausfallszenarien gehören die folgenden:

- Ein Beendigungscode ungleich Null von einem Container-Auftrag

- Ausfall oder Kündigung der EC2 Amazon-Instance
- Interner AWS Servicefehler oder Ausfall

Wenn ein Job an eine Auftragswarteschlange weitergeleitet und in den `RUNNING` Status versetzt wird, der als Versuch gewertet wird. Standardmäßig erhält jeder Auftrag einen Versuch, um entweder in den Auftragsstatus `SUCCEEDED` oder `FAILED` verschoben zu werden. Sowohl die Workflows für die Auftragsdefinition als auch für die Auftragsübermittlung können jedoch verwendet werden, um eine Wiederholungsstrategie mit einem bis zehn Versuchen festzulegen. Wenn evaluateOnExitangegeben, kann sie bis zu 5 Wiederholungsstrategien enthalten. Wenn evaluateOnExitangegeben, aber keine der Wiederholungsstrategien übereinstimmt, wird der Job erneut versucht. Fügen Sie für Jobs, die nicht mit Exit übereinstimmen, einen letzten Eintrag hinzu, der aus einem beliebigen Grund beendet wird. Dieses `evaluateOnExit` Objekt hat beispielsweise zwei Einträge mit der Aktion von `RETRY` und einen letzten Eintrag mit der Aktion von`EXIT`.

```
"evaluateOnExit": [
    {
        "action": "RETRY",
        "onReason": "AGENT"
    },
    {
        "action": "RETRY",
        "onStatusReason": "Task failed to start"
    },
    {
        "action": "EXIT",
        "onReason": "*"
    }
]
```

Zur Laufzeit wird die Umgebungsvariable `AWS_BATCH_JOB_ATTEMPT` auf die Nummer des entsprechenden Auftragsversuchs des Containers festgelegt. Der erste Versuch ist nummeriert`1`, und die nachfolgenden Versuche sind in aufsteigender Reihenfolge (z. B. 2, 3, 4).

Nehmen wir beispielsweise an, dass ein Jobversuch aus irgendeinem Grund fehlschlägt und die in der Konfiguration für die Wiederholung angegebene Anzahl von Versuchen größer ist als die `AWS_BATCH_JOB_ATTEMPT` Anzahl. Dann wird der Job wieder in den `RUNNABLE` Status versetzt. Weitere Informationen finden Sie unter Auftragsstatus.

> **Note**
>
> Aufträge, die storniert oder beendet wurden, werden nicht erneut versucht. Außerdem werden Jobs, die aufgrund einer ungültigen Jobdefinition fehlschlagen, nicht erneut versucht.

Weitere Informationen finden Sie unter Strategie erneut versuchenErstellen Sie eine Auftragsdefinition mit einem Knoten , Tutorial: Einen Job einreichen und Fehlercodes für gestoppte Aufgaben.

Abhängigkeiten von Job

Wenn Sie einen AWS Batch Job einreichen, können Sie den Job angeben IDs , von dem der Job abhängt. Wenn Sie dies tun, stellt der AWS Batch Scheduler sicher, dass Ihr Job erst ausgeführt wird, nachdem die angegebenen Abhängigkeiten erfolgreich abgeschlossen wurden. Nachdem diese erfolgreich waren, wechselt der abhängige Auftrag von `PENDING` zu `RUNNABLE` und dann zu `STARTING` und `RUNNING`. Wenn eine der Auftragsabhängigkeiten fehlschlägt, wechselt der abhängige Auftrag automatisch von `PENDING` zu `FAILED`.

Beispielsweise kann Auftrag A eine Abhängigkeit von bis zu 20 anderen Aufträge haben. Diese müssen erfolgreich sein, bevor er ausgeführt werden kann. Sie können dann weitere von Auftrag A und bis zu 19 weiteren Aufträgen abhängige Aufträge senden.

Bei Array-Aufträgen können Sie eine `SEQUENTIAL`-Typabhängigkeit angeben, ohne eine Auftrags-ID anzugeben, sodass jeder untergeordnete Array-Auftrag sequentiell abgeschlossen wird (beginnend mit Index 0). Sie können auch eine `N_TO_N`-Typabhängigkeit mit einer Auftrags-ID festlegen. So muss jeder untergeordnete Index dieses Auftrags warten, bis der entsprechende untergeordnete Index jeder Abhängigkeit abgeschlossen ist. Weitere Informationen finden Sie unter Ordnen Sie Jobs an.

Informationen zum Einreichen eines AWS Batch Jobs mit Abhängigkeiten finden Sie unterTutorial: Einen Job einreichen.

Ressourcenbewusste Planung ermöglicht es Ihnen, Jobs auf der Grundlage verbrauchbarer Ressourcen zu planen, die für die Ausführung Ihrer Jobs benötigt werden. Sie geben die verbrauchbaren Ressourcen an, die für die Ausführung eines Jobs benötigt werden, und Batch berücksichtigt diese Ressourcenabhängigkeiten bei der Planung eines Jobs. Sie können die Unterauslastung der Rechenressourcen verringern, indem Sie nur die Jobs zuweisen, für die alle

erforderlichen Ressourcen verfügbar sind. Die ressourcenorientierte Planung ist sowohl für FIFO- als auch für Fair-Share-Scheduling-Richtlinien verfügbar und kann mit allen von Batch unterstützten Computerplattformen verwendet werden, einschließlich EKS, ECS und Fargate. Es kann mit Array-Jobs, Multi-Node-Parallel-Jobs (MNP) und mit regulären Batch-Jobs verwendet werden.

Timeouts bei der Job

Sie können ein Timeout für Ihre Aufträge so konfigurieren, dass AWS Batch den Auftrag beendet, wenn die Ausführung länger als den angegebenen Wert dauert. Sie haben beispielsweise einen Auftrag, der zu seiner Ausführung Ihres Wissens nach nur 15 Minuten in Anspruch nehmen sollte. Manchmal bleibt Ihre Anwendung in einer Schleife hängen und wird endlos ausgeführt. Sie können also eine Zeitbeschränkung von 30 Minuten festlegen, bis der hängen gebliebene Auftrag beendet wird.

> **Important**
>
> Standardmäßig AWS Batch gibt es kein Job-Timeout. Wenn Sie kein Job-Timeout definieren, wird der Job ausgeführt, bis der Container beendet wird.

Sie geben entweder in Ihrer Auftragsdefinition oder bei der Übermittlung des Auftrags einen `attemptDurationSeconds`-Parameter an, der einen Mindestwert von 60 Sekunden haben muss. Wenn diese Anzahl von Sekunden nach dem `startedAt` Zeitstempel des Auftragsversuchs vergangen ist, wird der Job AWS Batch beendet. Auf der Datenverarbeitungsressource empfängt Ihr Auftrags-Container ein `SIGTERM`-Signal, um Ihrer Anwendung die Möglichkeit zu geben, ordnungsgemäß zu schließen. Wenn sich der Container nach 30 Sekunden noch immer in der Ausführung befindet, wird ein `SIGKILL`-Signal gesendet, um den Container zwangsweise herunterzufahren.

Timeout-Beendigungen werden auf einer Best-Effort-Basis abgewickelt. Sie sollten nicht erwarten, dass Ihr Timeout genau dann beendet wird, wenn das Zeitlimit für den Jobversuch überschritten wird (es kann einige Sekunden länger dauern). Wenn Ihre Anwendung eine genaue Timeout-Ausführung benötigt, sollten Sie diese Logik innerhalb der Anwendung implementieren. Wenn Sie eine große Anzahl von Aufträgen haben, bei denen gleichzeitig Zeitüberschreitungen auftreten, werden die Timeout-Beendigungen in einer FIFO(First-in-First-out)-Warteschlange abgearbeitet, wobei Aufträge stapelweise beendet werden.

> **Note**
>
> Es gibt keinen maximalen Timeout-Wert für einen AWS Batch Job.

Wenn ein Job wegen Überschreitung der Timeoutdauer beendet wird, wird er nicht erneut versucht. Wenn ein Auftragsversuch fehlschlägt, kann er wiederholt werden, wenn Wiederholversuche aktiviert sind, und der Zeitüberschreitungs-Countdown wird für den neuen Versuch erneut gestartet.

> **Important**
>
> Jobs, die auf Fargate-Ressourcen ausgeführt werden, können nicht damit rechnen, länger als 14 Tage zu laufen. Wenn die Timeout-Dauer 14 Tage überschreitet, sind die Fargate-Ressourcen möglicherweise nicht mehr verfügbar und der Job wird beendet.

Bei Array-Aufgaben haben untergeordnete Aufträge dieselbe Timeout-Konfiguration wie der übergeordnete Auftrag.

Hinweise zum Einreichen eines AWS Batch Jobs mit einer Timeout-Konfiguration finden Sie unter. Tutorial: Einen Job einreichen

Stellenangebote bei Amazon EKS

Ein Job ist die kleinste Arbeitseinheit in AWS Batch. Ein AWS Batch Job auf Amazon EKS hat eine one-to-one Zuordnung zu einem Kubernetes Pod. Eine AWS Batch Jobdefinition ist eine Vorlage für einen AWS Batch Job. Wenn Sie einen AWS Batch Job einreichen, verweisen Sie auf eine Jobdefinition, geben eine Job-Warteschlange als Ziel an und geben einen Namen für einen Job an. In der Auftragsdefinition eines AWS Batch Jobs auf Amazon EKS definiert der Parameter eksProperties den Parametersatz, den ein Job AWS Batch auf Amazon EKS unterstützt. In einer SubmitJobAnfrage ermöglicht der eksPropertiesOverrideParameter das Überschreiben einiger gängiger Parameter. Auf diese Weise können Sie Vorlagen von Jobdefinitionen für mehrere Jobs verwenden. Wenn ein Auftrag an Ihren Amazon EKS-Cluster gesendet wird, AWS Batch wandelt er den Auftrag in einen `podspec` (`Kind: Pod`) um. Der `podspec` verwendet einige zusätzliche AWS Batch Parameter, um sicherzustellen, dass Jobs korrekt skaliert und geplant werden. AWS Batch kombiniert Labels und Taints, um sicherzustellen, dass Jobs nur auf AWS Batch verwalteten Knoten ausgeführt werden und dass andere Pods nicht auf diesen Knoten ausgeführt werden.

> **Important**
>
> - Wenn der `hostNetwork` Parameter in einer Amazon EKS-Auftragsdefinition nicht explizit festgelegt ist, wird für den Pod-Netzwerkmodus AWS Batch standardmäßig der Hostmodus verwendet. Insbesondere werden die folgenden Einstellungen angewendet: `hostNetwork=true` und`dnsPolicy=ClusterFirstWithHostNet`.
> - AWS Batch bereinigt Job-Pods, kurz nachdem ein Pod seinen Job abgeschlossen hat. Um Pod-Anwendungsprotokolle zu sehen, konfigurieren Sie einen Protokollierungsdienst für Ihren Cluster. Weitere Informationen finden Sie unter Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen.

Themen

- Tutorial: Ordnen Sie einen laufenden Job einem Pod und einem Knoten zu
- Tutorial: Ordnen Sie einen laufenden Pod seinem Job zu

Tutorial: Ordnen Sie einen laufenden Job einem Pod und einem Knoten zu

Für einen laufenden Job wurden die `podName` `nodeName` Parameter für den aktuellen Jobversuch festgelegt. `podProperties` Verwenden Sie den DescribeJobsAPI-Vorgang, um diese Parameter anzuzeigen.

Es folgt eine Beispielausgabe.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
 "jobs": [
  {
   "status": "RUNNING",
   "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-
f2baf7e51b04",
   "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/
MyJobOnEks_SleepWithRequestsOnly:1",
   "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
   "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
   "eksProperties": {
    "podProperties": {
     "nodeName": "ip-192-168-55-175.ec2.internal",
```

```
      "containers": [
       {
        "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
        "resources": {
         "requests": {
          "cpu": "1",
          "memory": "1024Mi"
         }
        }
       }
      ],
      "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
     }
    }
   }
  ]
}
```

Bei einem Job mit aktivierten Wiederholungsversuchen steht das `podName` `nodeName` Ende jedes abgeschlossenen Versuchs im `eksAttempts` Listenparameter des DescribeJobsAPI-Vorgangs. Das `podName` Ende `nodeName` des aktuell ausgeführten Versuchs befindet sich im `podProperties` Objekt.

Tutorial: Ordnen Sie einen laufenden Pod seinem Job zu

Ein Pod hat Beschriftungen, die das `jobId` `uuid` Ende der Rechenumgebung angeben, zu der er gehört. AWS Batch fügt Umgebungsvariablen ein, sodass die Laufzeit des Jobs auf Jobinformationen verweisen kann. Weitere Informationen finden Sie unter AWS Batch Variablen der Arbeitsumgebung. Sie können diese Informationen anzeigen, indem Sie den folgenden Befehl ausführen. Die Ausgabe sieht wie folgt aus.

```
$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-
namespace
Name:         aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:    my-aws-batch-namespace
Priority:     0
Node:         ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:   Wed, 26 Oct 2022 00:30:48 +0000
Labels:       batch.amazonaws.com/compute-environment-uuid=5c19160b-
d450-31c9-8454-86cf5b30548f
              batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
              batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
```

```
...
Status:       Running
IP:           192.168.45.88
IPs:
  IP:  192.168.45.88
Containers:
  default:
    Image:         public.ecr.aws/amazonlinux/amazonlinux:2
    ...
    Environment:
      AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647
      AWS_BATCH_JOB_ID:                   f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
      AWS_BATCH_JQ_NAME:                  My-Eks-JQ1
      AWS_BATCH_JOB_ATTEMPT:              1
      AWS_BATCH_CE_NAME:                  My-Eks-CE1

...
```

Funktionen, die AWS Batch Amazon EKS-Jobs unterstützen

Dies sind die AWS Batch spezifischen Funktionen, die auch für Kubernetes Jobs gelten, die auf Amazon EKS ausgeführt werden:

- Abhängigkeiten von Job
- Ordnen Sie Jobs an
- Timeouts bei der Job
- Automatisierte Auftragswiederholungen
- Verwenden Sie Fair-Share-Scheduling, um Jobs besser planen zu können

Kubernetes**`Secrets`** und **`ServiceAccounts`**

AWS Batch unterstützt Referenzierung Kubernetes `Secrets` und`ServiceAccounts`. Sie können Pods so konfigurieren, dass sie Amazon EKS-IAM-Rollen für Dienstkonten verwenden. Weitere Informationen finden Sie unter Konfiguration von Pods für die Verwendung eines Kubernetes Servicekontos im Amazon EKS-Benutzerhandbuch.

Verwandte Dokumente

- Überlegungen zu Arbeitsspeicher und vCPU für AWS Batch Amazon EKS
- Führen Sie GPU-Jobs aus

- Jobs, die in einem `RUNNABLE` Status feststecken

parallel Jobs mit mehreren Knoten

Sie können parallel Jobs mit mehreren Knoten verwenden, um einzelne Jobs auszuführen, die sich über mehrere EC2 Amazon-Instances erstrecken. Mit parallel Jobs mit AWS Batch mehreren Knoten (auch bekannt als Gruppenplanung) können Sie umfangreiche Hochleistungsrechneranwendungen und das Training verteilter GPU-Modelle ausführen, ohne EC2 Amazon-Ressourcen direkt starten, konfigurieren und verwalten zu müssen. Ein parallel Job mit AWS Batch mehreren Knoten ist mit jedem Framework kompatibel, das IP-basierte Kommunikation zwischen Knoten unterstützt. Beispiele hierfür sind Apache MXNet TensorFlow, Caffe2 oder Message Passing Interface (MPI).

Parallele Aufträge mit mehreren Knoten werden als ein einzelner Auftrag übermittelt. Ihre Auftragsdefinition (oder Überschreibungen durch den Auftragsübergabeknoten) legt jedoch die Anzahl der Knoten fest, die für den Auftrag erstellt werden müssen, und welche Knotengruppen erstellt werden müssen. Jeder parallele Auftrag mit mehreren Knoten enthält einen Hauptknoten, der zuerst gestartet wird. Sobald der Hauptknoten aktiv ist, werden die untergeordneten Knoten gestartet. Der Job ist nur abgeschlossen, wenn der Hauptknoten beendet wird. Alle untergeordneten Knoten werden dann gestoppt. Weitere Informationen finden Sie unter Knotengruppen.

parallel Jobknoten mit mehreren Knoten sind Single-Tenants. Das bedeutet, dass auf jeder EC2 Amazon-Instance nur ein einziger Job-Container ausgeführt wird.

Der letzte Auftragsstatus (`SUCCEEDED` oder `FAILED`) wird durch den letzten Auftragsstatus des Hauptknotens bestimmt. Um den Status eines parallel Jobs mit mehreren Knoten abzurufen, beschreiben Sie den Job anhand der Job-ID, die beim Absenden des Jobs zurückgegeben wurde. Wenn Sie die Details für untergeordnete Knoten benötigen, beschreiben Sie jeden untergeordneten Knoten einzeln. Sie können Knoten anhand der `#N` Notation adressieren (beginnend mit 0). Um beispielsweise auf die Details des zweiten Knotens eines Jobs zuzugreifen, beschreiben Sie *`aws_batch_job_id`* #1 mithilfe der AWS Batch DescribeJobsAPI-Operation. Die Informationen `started`, `stoppedAt`, `statusReason` und `exit` für einen parallelen Auftrag mit mehreren Knoten stammen aus dem Hauptknoten.

Wenn Sie Auftragswiederholungen angeben, führt ein Ausfall des Hauptknotens zu einem erneuten Versuch. Ausfälle von untergeordneten Knoten führen nicht zu weiteren Versuchen. Jeder neue Versuch eines parallelen Auftrags mit mehreren Knoten aktualisiert den entsprechenden Versuch seiner zugehörigen untergeordneten Knoten.

Um parallel Jobs mit mehreren Knoten ausführen zu können AWS Batch, muss Ihr Anwendungscode die Frameworks und Bibliotheken enthalten, die für die verteilte Kommunikation erforderlich sind.

Themen

- Umgebungsvariablen
- Knotengruppen
- Joblebenszyklus für MNP-Jobs
- Überlegungen zur Rechenumgebung für MNP mit AWS Batch

Umgebungsvariablen

Zur Laufzeit werden für jeden Knoten die Standardumgebungsvariablen konfiguriert, die alle AWS Batch Jobs erhalten. Darüber hinaus sind die Knoten mit den folgenden Umgebungsvariablen konfiguriert, die für parallel Jobs mit mehreren Knoten spezifisch sind:

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Diese Variable ist auf die Indexnummer des Hauptknotens des Auftrags festgelegt. Ihr Anwendungscode kann den mit dem `AWS_BATCH_JOB_MAIN_NODE_INDEX` `AWS_BATCH_JOB_NODE_INDEX` auf einem einzelnen Knoten vergleichen, um festzustellen, ob es sich um den Hauptknoten handelt.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Diese Variable wird nur in untergeordneten Knoten für parallel Jobs mit mehreren Knoten gesetzt. Diese Variable ist auf dem Hauptknoten nicht vorhanden. Diese Variable ist auf die private IPv4 Adresse des Hauptknotens des Jobs gesetzt. Der Anwendungscode Ihres untergeordneten Knotens kann diese Adresse zur Kommunikation mit dem Hauptknoten verwenden.

`AWS_BATCH_JOB_NODE_INDEX`

Diese Variable ist auf die Indexnummer des Knotens festgelegt. Der Knotenindex beginnt bei 0 und jeder Knoten erhält eine eindeutige Indexnummer. Beispielsweise weist ein paralleler Auftrag mit mehreren Knoten und 10 untergeordneten Aufträgen die Indexwerte 0–9 auf.

`AWS_BATCH_JOB_NUM_NODES`

Diese Variable ist auf die Anzahl der Knoten festgelegt, die Sie für Ihren parallelen Auftrag mit mehreren Knoten angefordert haben.

Knotengruppen

Eine Knotengruppe ist eine identische Gruppe von Job-Knoten, die alle dieselben Container-Eigenschaften haben. Sie können AWS Batch damit bis zu fünf verschiedene Knotengruppen für jeden Job angeben.

Jede Gruppe kann ihre eigenen Container-Images, Befehle, Umgebungsvariablen und so weiter besitzen. Sie können beispielsweise einen Job einreichen, für den eine einzelne `c5.xlarge` Instanz für den Hauptknoten und fünf untergeordnete `c5.xlarge` Instanzknoten erforderlich sind. Jede dieser unterschiedlichen Knotengruppen kann unterschiedliche Container-Images oder Befehle angeben, die für jeden Job ausgeführt werden sollen.

Alternativ können alle Knoten in Ihrem Job eine einzelne Knotengruppe verwenden. Darüber hinaus kann Ihr Anwendungscode zwischen Knotenrollen wie dem Hauptknoten und dem untergeordneten Knoten unterscheiden. Dazu wird die `AWS_BATCH_JOB_MAIN_NODE_INDEX` Umgebungsvariable mit ihrem eigenen Wert für verglichen`AWS_BATCH_JOB_NODE_INDEX`. Sie können bis zu 1.000 Knoten in einem einzigen Job haben. Dies ist das Standardlimit für Instances in einem Amazon ECS-Cluster. Sie können eine Erhöhung dieses Limits beantragen.

> **Note**
>
> Derzeit müssen alle Knotengruppen in einem parallelen Auftrag mit mehreren Knoten den gleichen Instance-Typ verwenden.

Joblebenszyklus für MNP-Jobs

Wenn Sie einen parallel Job mit mehreren Knoten einreichen, erhält der Job den `SUBMITTED` Status. Anschließend wartet der Job darauf, dass alle Auftragsabhängigkeiten abgeschlossen sind. Der Job wird ebenfalls in den `RUNNABLE` Status versetzt. Schließlich wird die AWS Batch Instance-Kapazität bereitgestellt, die für die Ausführung Ihres Jobs erforderlich ist, und diese Instances werden gestartet.

Jeder parallele Auftrag mit mehreren Knoten enthält einen Hauptknoten. Der Hauptknoten ist eine einzelne Unteraufgabe, die das Ergebnis des eingereichten Auftrags mit mehreren Knoten AWS Batch überwacht und bestimmt. Der Hauptknoten wird zum ersten Mal gestartet und wechselt in den `STARTING`-Status. Der im `attemptDurationSeconds` Parameter angegebene Timeout-Wert gilt für den gesamten Job und nicht für die Knoten.

Wenn der Hauptknoten den `RUNNING` Status erreicht, nachdem der Container des Knotens ausgeführt wurde, werden die untergeordneten Knoten gestartet und nehmen ebenfalls diesen `STARTING` Status an. Die untergeordneten Knoten werden in zufälliger Reihenfolge angezeigt. Es gibt keine Garantien hinsichtlich des Zeitpunkts oder der Reihenfolge, in der die untergeordneten Knoten starten. Um sicherzustellen, dass sich alle Knoten der Jobs im `RUNNING` Status befinden, nachdem der Container des Knotens ausgeführt wurde, kann Ihr Anwendungscode die AWS Batch API abfragen, um Informationen über den Hauptknoten und den untergeordneten Knoten abzurufen. Alternativ kann der Anwendungscode warten, bis alle Knoten online sind, bevor eine verteilte Verarbeitungsaufgabe gestartet wird. Die private IP-Adresse des Hauptknotens ist als `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`-Umgebungsvariable in jedem untergeordneten Knoten verfügbar. Ihr Anwendungscode kann diese Informationen verwenden, um Daten und die Kommunikation zwischen den einzelnen Aufgaben zu koordinieren.

Wenn einzelne Knoten beendet werden, wechseln sie je nach ihrem Beendigungscode zu `SUCCEEDED` oder `FAILED`. Wenn der Hauptknoten beendet wird, gilt der Auftrag als abgeschlossen und alle seine untergeordneten Knoten werden beendet. Wenn ein untergeordneter Knoten ausfällt, werden AWS Batch keine Aktionen für die anderen Knoten im Job ausgeführt. Wenn Sie nicht möchten, dass Ihr Job mit einer reduzierten Anzahl von Knoten fortgeführt wird, müssen Sie dies in Ihrem Anwendungscode berücksichtigen. Dadurch wird der Job beendet oder abgebrochen.

Überlegungen zur Rechenumgebung für MNP mit AWS Batch

Es gibt mehrere Dinge, die Sie berücksichtigen sollten, wenn Sie die Datenverarbeitungsumgebungen so konfigurieren, dass parallele Aufträge mit mehreren Knoten mit AWS Batch ausgeführt werden.

- parallel Jobs mit mehreren Knoten werden in `UNMANAGED` Computerumgebungen nicht unterstützt.
- Wenn Sie parallel Jobs mit mehreren Knoten an eine Rechenumgebung senden möchten, erstellen Sie eine Cluster-Platzierungsgruppe in einer einzigen Availability Zone und ordnen Sie sie Ihren Rechenressourcen zu. Dadurch bleiben Ihre parallel Jobs mit mehreren Knoten in einer logischen Gruppierung von Instanzen mit hohem Netzwerkflusspotenzial nahe beieinander. Weitere Informationen finden Sie unter Placement-Gruppen im EC2 Amazon-Benutzerhandbuch.
- parallel Jobs mit mehreren Knoten werden in Computerumgebungen, die Spot-Instances verwenden, nicht unterstützt.
- AWS Batch parallel Jobs mit mehreren Knoten verwenden den Amazon `awsvpc` ECS-Netzwerkmodus, der Ihren parallel Job-Containern mit mehreren Knoten dieselben Netzwerkeigenschaften wie EC2 Amazon-Instances verleiht. Jeder Container eines parallelen

Auftrags mit mehreren Knoten erhält seine eigene Elastic Network-Schnittstelle, eine primäre private IP-Adresse und einen internen DNS-Hostnamen. Die Netzwerkschnittstelle wird im gleichen VPC-Subnetz erstellt wie ihre Host-Datenverarbeitungsressource.

- Ihrer Computerumgebung sind möglicherweise nicht mehr als fünf Sicherheitsgruppen zugeordnet. Die Elastic Network-Schnittstellen, die erstellt und an eine MNP-Aufgabe angehängt werden, verwenden die in Ihrer Computerumgebung angegebenen Sicherheitsgruppen. Wenn Sie keine Sicherheitsgruppe angeben, wird die Standardsicherheitsgruppe für die VPC verwendet.
- Der `awsvpc` Netzwerkmodus bietet keine elastischen Netzwerkschnittstellen für parallel Jobs mit mehreren Knoten und öffentlichen IP-Adressen. Um auf das Internet zuzugreifen, müssen Ihre Datenverarbeitungsressourcen in einem privaten Subnetz gestartet werden, das für die Verwendung eines NAT-Gateways konfiguriert ist. Weitere Informationen finden Sie unter NAT-Gateways im Amazon VPC-Benutzerhandbuch. Bei der Kommunikation zwischen den Knoten muss die private IP-Adresse oder der DNS-Hostname für den Knoten verwendet werden. parallel Jobs mit mehreren Knoten, die auf Rechenressourcen in öffentlichen Subnetzen ausgeführt werden, haben keinen ausgehenden Netzwerkzugriff. Weitere Informationen zum Erstellen einer VPC mit privaten Subnetzen und einem NAT-Gateway finden Sie unter Erstellen einer Virtual Private Cloud .
- Die Elastic Network-Schnittstellen, die erstellt und an Ihre Rechenressourcen angehängt werden, können nicht manuell getrennt oder von Ihrem Konto geändert werden. Dadurch soll verhindert werden, dass versehentlich eine elastic network interface gelöscht wird, die mit einem laufenden Job verknüpft ist. Um die Elastic Network-Schnittstellen für eine Aufgabe freizugeben, beenden Sie den Auftrag.
- Ihre Rechenumgebung muss über genügend maximale V verfügenCPUs , um Ihren parallel Job mit mehreren Knoten zu unterstützen.
- Ihr EC2 Amazon-Instance-Kontingent beinhaltet die Anzahl der Instances, die für die Ausführung Ihres Jobs erforderlich sind. Nehmen wir zum Beispiel an, dass für Ihren Job 30 Instances erforderlich sind, Ihr Konto jedoch nur 20 Instances in einer Region ausführen kann. Dann bleibt Ihr Job im `RUNNABLE` Status hängen.
- Wenn Sie in einem parallel Job mit mehreren Knoten einen Instanztyp für eine Knotengruppe angeben, muss Ihre Rechenumgebung diesen Instanztyp starten.

parallel Jobs mit mehreren Knoten auf Amazon EKS

Sie können AWS Batch Amazon Elastic Kubernetes Service verwenden, um parallel Multi-Node-Jobs (MNP) (auch bekannt als Gang Scheduling) auf Ihren verwalteten Clustern auszuführen. Kubernetes

Diese Option wird häufig für große, eng gekoppelte Hochleistungsjobs verwendet, die nicht auf einer einzigen Amazon Elastic Compute Cloud-Instance ausgeführt werden können. Weitere Informationen finden Sie unter parallel Jobs mit mehreren Knoten.

Sie können diese Funktion verwenden, um von Amazon EKS verwaltete Kubernetes spezifische Hochleistungscomputeranwendungen, das Training großer Sprachmodelle und andere Jobs im Bereich künstliche Intelligenz (KI) /maschinelles Machine Learning (ML) auszuführen.

Themen

- MNP-Jobs ausführen
- Erstellen Sie eine Amazon EKS MNP-Jobdefinition
- Einen Amazon EKS MNP-Job einreichen
- Eine Amazon EKS MNP-Auftragsdefinition überschreiben

MNP-Jobs ausführen

AWS Batch unterstützt MNP-Jobs auf Amazon Elastic Container Service und Amazon EKS mithilfe von Amazon EC2. Im Folgenden finden Sie weitere Einzelheiten zu den Instance- und Container-Parametern für die Funktion.

Instanzkontingente für MNP auf Amazon EKS

- Bis zu 1000 Instances können für einen einzelnen MNP-Job verwendet werden.
- Bis zu 5000 Instances können einem einzelnen Amazon EKS-Cluster beitreten.
- Bis zu 5 Computerumgebungen können geclustert und an eine Job-Warteschlange angehängt werden.

Sie können beispielsweise bis zu 5 geclusterte Rechenumgebungen in einer Job-Warteschlange und 1000 Instanzen in jeder Rechenumgebung skalieren.

Zusätzlich zu den Instanzparametern ist es wichtig zu beachten, dass Sie Fargate über keinen der beiden Dienste für MNP-Jobs verwenden können.

Sie können in jedem MNP-Job nur einen Instanztyp verwenden. Sie können den Instanztyp ändern, indem Sie die Rechenumgebung aktualisieren oder wenn Sie eine neue Rechenumgebung definieren. Sie können auch den Instanztyp angeben und vCPU- und Speicheranforderungen angeben, wenn Sie die Jobdefinition erstellen.

Containerkontingente für MNP auf Amazon EKS

- Ein parallel Job mit mehreren Knoten unterstützt einen Pod pro Knoten.
- Bis zu 10 Container (oder 10 Init-Container). Weitere Informationen finden Sie unter Init-Container in der Kubernetes-Dokumentation.) in jedem Pod.
- Bis zu 5 Knotenbereiche in jedem MNP-Job.
- Bis zu 10 verschiedene Container-Images in jedem Knotenbereich.

Sie können beispielsweise bis zu 10.000 Container in einem einzigen MNP-Job ausführen, der 5 Knotenbereiche und insgesamt 50 eindeutige Bilder enthält.

Ausführung von MNP-Jobs in einer privaten Amazon VPC und einem Amazon EKS-Cluster

MNP-Jobs können auf jedem Amazon EKS-Cluster ausgeführt werden, unabhängig davon, ob er über ein öffentliches Internet verfügt oder nicht. Wenn Sie einen Amazon EKS-Cluster mit ausschließlich privatem Netzwerkzugriff verwenden, stellen Sie sicher, dass dieser auf die Amazon EKS-Steuerebene und den verwalteten Kubernetes API-Server zugreifen AWS Batch kann. Sie können den erforderlichen Zugriff über Amazon Virtual Private Cloud Cloud-Endpunkte gewähren. Weitere Informationen finden Sie unter Konfiguration eines Endpunktdienstes.

Amazon EKS-Cluster-Pods können kein Image von einer öffentlichen Quelle herunterladen, da die private VPC keinen Internetzugang hat. Ihr Amazon EKS-Cluster muss Images aus einer Container-Registry abrufen, die sich in Ihrer Amazon VPC befindet. Sie können eine Amazon Elastic Container Registry (Amazon ECR) in Ihrer Amazon VPC erstellen und Container-Images dorthin kopieren, damit Ihre Knoten darauf zugreifen können.

Sie können mit Amazon ECR auch eine Pull-Through-Cache-Regel erstellen. Sobald eine Pull-Through-Cache-Regel für eine externe öffentliche Registrierung erstellt wurde, können Sie einfach ein Bild aus dieser externen öffentlichen Registrierung mit Ihrer privaten Amazon ECR-Registrierungs-URI abrufen. Dann erstellt Amazon ECR ein Repository und speichert das Bild im Cache. Wenn ein zwischengespeichertes Image mithilfe des privaten Amazon-ECR-Registrierung-URI abgerufen wird, überprüft Amazon ECR die Remote-Registrierung, um festzustellen, ob eine neue Version des Images vorhanden ist, und aktualisiert Ihre private Registrierung bis zu einmal alle 24 Stunden. Weitere Informationen finden Sie unter Erstellen einer Pull-Through-Cache-Regel in Amazon ECR.

Benachrichtigung über einen Fehler

Wenn Ihre MNP-Jobs blockiert sind, können Sie Benachrichtigungen über The AWS Management Console und Amazon EventBridge erhalten. Wenn beispielsweise ein MNP-Job an der Spitze der Warteschlange feststeckt, können Sie über das Problem sowie Informationen zur Ursache des Problems informiert werden, sodass Sie umgehend Maßnahmen ergreifen können, um Ihre Job-Warteschlange zu entsperren. Optional können Sie den MNP-Job automatisch beenden, wenn innerhalb eines bestimmten Zeitraums, der in der Vorlage für die Auftragswarteschlange definiert werden kann, keine Aktion ausgeführt wird. Weitere Informationen finden Sie unter Ressource: Blockierte Ereignisse in der Jobwarteschlange.

Erstellen Sie eine Amazon EKS MNP-Jobdefinition

Um MNP-Jobs auf Amazon EKS zu definieren und auszuführen, gibt es neue Parameter innerhalb der `RegisterJobDefinition`und `SubmitJob`API-Operationen.

- Verwenden Sie den `eksPropertiesnodeProperties`Abschnitt unten, um Ihre MNP-Jobdefinition zu definieren.
- Verwenden Sie die `eksPropertiesOverride`Option unter dem `nodePropertyOverrides`Abschnitt, um die in der Jobdefinition definierten Parameter zu überschreiben, wenn Sie einen MNP-Job einreichen.

Diese Aktionen können durch API-Operationen und die AWS Management Console definiert werden.

Referenz: Registrieren Sie die Payload für die Amazon EKS MNP-Jobdefinitionsanforderung

Das folgende Beispiel zeigt, wie Sie eine Amazon EKS MNP-Auftragsdefinition mit zwei Knoten registrieren können.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes" : "0:",
```

```
        "eksProperties": {
          "podProperties": {
            "containers": [
              {
                "name": "test-eks-container-1",
                "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                "command": [
                  "sleep",
                  "60"
                ],
                "resources": {
                  "limits": {
                    "cpu": "1",
                    "memory": "1024Mi"
                  }
                },
                "securityContext":{
                  "runAsUser":1000,
                  "runAsGroup":3000,
                  "privileged":true,
                  "readOnlyRootFilesystem":true,
                  "runAsNonRoot":true
               }
              }
            ],
            "initContainers": [
               {
                  "name":"init-ekscontainer",
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                  "command": [
                     "echo",
                     "helloWorld"
                   ],
                   "resources": {
                     "limits": {
                       "cpu": "1",
                       "memory": "1024Mi"
                     }
                  }
               }
            ],
            "metadata": {
               "labels": {
                  "environment" : "test"
```

```
                }
            }
          }
        }
      }
    ]
  }
}
```

Um die Auftragsdefinition mithilfe von zu registrieren AWS CLI, kopieren Sie die Definition in eine lokale Datei mit dem Namen MyEksMnpJobDefinition.json und führen Sie den folgenden Befehl aus.

```
aws batch register-job-definition --cli-input-json file://MyEksMnpJobDefinition.json
```

Sie erhalten die folgende JSON-Antwort.

```
{
    "jobDefinitionName": "MyEksMnpJobDefinition",
    "jobDefinitionArn": "arn:aws:batch:us-east-1:0123456789:job-definition/
MyEksMnpJobDefinition:1",
    "revision": 1
}
```

Einen Amazon EKS MNP-Job einreichen

Um einen Job mit der registrierten Jobdefinition einzureichen, geben Sie den folgenden Befehl ein. Ersetzen Sie den Wert von durch <EKS_JOB_QUEUE_NAME>den Namen oder ARN einer bereits vorhandenen Auftragswarteschlange, die mit einer Amazon EKS-Rechenumgebung verknüpft ist.

```
aws batch submit-job --job-queue <EKS_JOB_QUEUE_NAME> \
    --job-definition MyEksMnpJobDefinition \
    --job-name myFirstEksMnpJob
```

Sie erhalten die folgende JSON-Antwort.

```
{
    "jobArn": "arn:aws:batch:region:account:job/9b979cce-9da0-446d-90e2-ffa16d52af68",
    "jobName": "myFirstEksMnpJob",
    "jobId": "<JOB_ID>"
```

```
}
```

Sie können den Status des Jobs anhand der zurückgegebenen jobId mit dem folgenden Befehl überprüfen.

```
aws batch describe-jobs --jobs <JOB_ID>
```

Eine Amazon EKS MNP-Auftragsdefinition überschreiben

Optional können Sie die Details der Auftragsdefinition überschreiben (z. B. die Änderung der MNP-Auftragsgröße oder der Details des untergeordneten Auftrags). Im Folgenden finden Sie ein Beispiel für eine JSON-Anforderungs-Payload zum Senden eines MNP-Jobs mit fünf Knoten sowie Änderungen am Befehl des `test-eks-container-1` Containers.

```
{
  "numNodes": 5,
  "nodePropertyOverrides": [
    {
      "targetNodes": "0:",
      "eksPropertiesOverride": {
        "podProperties": {
          "containers": [
            {
              "name": "test-eks-container-1",
              "command": [
                "sleep",
                "150"
              ]
            }
          ]
        }
      }
    }
  ]
}
```

Um einen Job mit diesen Überschreibungen einzureichen, speichern Sie das Beispiel in einer lokalen Datei, eks-mnp-job-nodeoverride.json, und verwenden Sie die, um den Job mit den Überschreibungen AWS CLI zu senden.

Ordnen Sie Jobs an

Ein Array-Job ist ein Job, der gemeinsame Parameter wie JobdefinitionCPUs, V und Speicher verwendet. Er wird als Sammlung verwandter, aber separater Basisjobs ausgeführt, die möglicherweise auf mehrere Hosts verteilt sind und gleichzeitig ausgeführt werden. Array-Jobs sind die effizienteste Methode, um extrem parallel Jobs wie Monte-Carlo-Simulationen, parametrische Sweeps oder umfangreiche Rendering-Jobs auszuführen.

AWS Batch Array-Jobs werden genauso wie normale Jobs eingereicht. Sie geben jedoch eine Array-Größe (zwischen 2 und 10.000) an. Diese legt fest, wie viele untergeordnete Aufträge im Array ausgeführt werden sollen. Wenn Sie einen Auftrag mit einer Array-Größe von 1000 senden, wird ein einzelner Auftrag ausgeführt und erzeugt 1000 untergeordnete Aufträge. Der Array-Auftrag ist eine Referenz oder ein Zeiger, um alle untergeordneten Aufträge zu verwalten. Auf diese Weise können Sie große Workloads mit einer einzigen Abfrage einreichen. Das im `attemptDurationSeconds` Parameter angegebene Timeout gilt für jeden untergeordneten Job. Der übergeordnete Array-Job hat kein Timeout.

Wenn Sie einen Array-Job einreichen, erhält der übergeordnete Array-Job eine normale AWS Batch Job-ID. Jeder untergeordnete Job hat dieselbe Basis-ID. Der Array-Index für den untergeordneten Job wird jedoch an das Ende der übergeordneten ID angehängt, z. B. *`example_job_ID`*`:0` für den ersten untergeordneten Job des Arrays.

Der übergeordnete Array-Job kann den `SUCCEEDED` Status `SUBMITTEDPENDING`,`FAILED`, oder annehmen. Ein übergeordneter Array-Job wird aktualisiert`PENDING`, wenn ein untergeordneter Job aktualisiert wird`RUNNABLE`. Weitere Hinweise zu Auftragsabhängigkeiten finden Sie unterAbhängigkeiten von Job.

Zur Laufzeit wird die Umgebungsvariable `AWS_BATCH_JOB_ARRAY_INDEX` auf die Nummer des entsprechenden Auftrags-Array-Index des Containers festgelegt. Der erste Array-Auftragsindex ist nummeriert`0`, und die nachfolgenden Versuche erfolgen in aufsteigender Reihenfolge (z. B. 1, 2 und 3). Sie können diesen Indexwert dazu verwenden, zu steuern, wie Ihre untergeordneten Array-Auftragselemente differenziert werden. Weitere Informationen finden Sie unter Tutorial: Verwenden Sie den Array-Jobindex, um die Jobdifferenzierung zu steuern.

Für Array-Auftrags-Abhängigkeiten können Sie einen Typ für eine Abhängigkeit angeben (z. B. `SEQUENTIAL` oder `N_TO_N`). Sie können eine `SEQUENTIAL`-Typabhängigkeit angeben, ohne eine Auftrags-ID anzugeben, sodass jeder untergeordnete Array-Auftrag sequentiell abgeschlossen wird (beginnend mit Index 0). Wenn Sie z. B. einen Array-Auftrag mit einer Array-Größe von 100

senden und eine Abhängigkeit vom Typ `SEQUENTIAL` angeben, werden 100 untergeordnete Aufträge sequentiell erzeugt, wobei der erste untergeordnete Auftrag erfolgreich sein muss, bevor der nächste untergeordnete Auftrag startet. Die folgende Abbildung zeigt Auftrag A, einen Array-Auftrag mit einer Array-Größe von 10. Jeder Auftrag im untergeordneten Index von Auftrag A ist vom vorherigen untergeordneten Auftrag abhängig. Auftrag A:1 kann erst gestartet werden, wenn Auftrag A:0 beendet ist.

Sie können auch eine Abhängigkeit vom `N_TO_N`-Typ mit einer Aufgaben-ID für Array-Aufgaben angeben. So muss jeder untergeordnete Index dieses Auftrags warten, bis der entsprechende untergeordnete Index jeder Abhängigkeit abgeschlossen ist. Die folgende Abbildung zeigt Job A und Job B, zwei Array-Jobs mit einer Array-Größe von jeweils 10.000. Jeder Auftrag im untergeordneten Index von Auftrag B ist abhängig von dem entsprechenden Index in Auftrag A. Auftrag B:1 kann erst gestartet werden, wenn Auftrag A:1 beendet ist.

Wenn Sie einen übergeordneten Array-Auftrag abbrechen oder beenden, werden alle untergeordneten Jobs gleichzeitig storniert oder beendet. Sie können einzelne untergeordnete Jobs stornieren oder beenden (wodurch sie in einen `FAILED` Status versetzt werden), ohne dass sich dies auf die anderen untergeordneten Jobs auswirkt. Wenn jedoch ein untergeordneter Array-Auftrag fehlschlägt (eigenständig oder durch manuelles Abbrechen oder Beenden des Auftrags), schlägt auch der übergeordnete Job fehl. In diesem Szenario geht der übergeordnete Job zu dem `FAILED` Zeitpunkt über, zu dem alle untergeordneten Jobs abgeschlossen sind.

Themen

- Referenz: Beispiel für einen Array-Job-Workflow
- Tutorial: Verwenden Sie den Array-Jobindex, um die Jobdifferenzierung zu steuern

Referenz: Beispiel für einen Array-Job-Workflow

Ein üblicher Arbeitsablauf für AWS Batch Kunden besteht darin, einen vorausgesetzten Setup-Job auszuführen, eine Reihe von Befehlen für eine große Anzahl von Eingabeaufgaben auszuführen

und dann mit einem Job abzuschließen, der Ergebnisse aggregiert und Zusammenfassungsdaten in Amazon S3, DynamoDB, Amazon Redshift oder Aurora schreibt.

Zum Beispiel:

- `JobA`: Ein standardmäßiger Nicht-Array-Job, der eine schnelle Auflistung und Metadatenvalidierung von Objekten in einem Amazon S3 S3-Bucket durchführt,`BucketA`. Die SubmitJobJSON-Syntax lautet wie folgt.

```
{
    "jobName": "JobA",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobA-list-and-validate:1"
}
```

- `JobB`: Ein Array-Job mit 10.000 Kopien, der `JobA` davon abhängig ist und CPU-intensive Befehle für jedes Objekt ausführt `BucketA` und Ergebnisse in dieses hochlädt. `BucketB` Die SubmitJobJSON-Syntax lautet wie folgt.

```
{
    "jobName": "JobB",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobB-CPU-Intensive-Processing:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "4096"
            },
            {
                "type": "VCPU",
                "value": "32"
            }
        ]
   }
    "arrayProperties": {
        "size": 10000
    },
    "dependsOn": [
        {
            "jobId": "JobA_job_ID"
  }
```

```
    ]
}
```

- `JobC`: Ein weiterer Array-Job `JobB` mit 10.000 Kopien, von dem ein `N_TO_N` Abhängigkeitsmodell abhängig ist, das speicherintensive Befehle für jedes Element ausführt`BucketB`, Metadaten in DynamoDB schreibt und die resultierende Ausgabe dorthin hochlädt. `BucketC` Die SubmitJobJSON-Syntax lautet wie folgt.

```
{
    "jobName": "JobC",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobC-Memory-Intensive-Processing:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32768"
            },
            {
                "type": "VCPU",
                "value": "1"
            }
        ]
   }
    "arrayProperties": {
        "size": 10000
    },
    "dependsOn": [
        {
            "jobId": "JobB_job_ID",
            "type": "N_TO_N"
        }
    ]
}
```

- `JobD`: Ein Array-Job, der 10 Validierungsschritte ausführt, die jeweils DynamoDB abfragen müssen und mit jedem der oben genannten Amazon S3 S3-Buckets interagieren können. Jeder der Schritte in `JobD` führt denselben Befehl aus. Das Verhalten ist jedoch je nach Wert der `AWS_BATCH_JOB_ARRAY_INDEX` Umgebungsvariablen im Container des Jobs unterschiedlich. Diese Überprüfungsschritte werden sequentiell ausgeführt (z. B. `JobD:0` und dann`JobD:1`). Die SubmitJobJSON-Syntax lautet wie folgt.

```
{
    "jobName": "JobD",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobD-Sequential-Validation:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32768"
            },
            {
                "type": "VCPU",
                "value": "1"
            }
        ]
   }
    "arrayProperties": {
        "size": 10
    },
    "dependsOn": [
        {
            "jobId": "JobC_job_ID"
        },
        {
            "type": "SEQUENTIAL"
        },

    ]
}
```

- `JobE`: Ein letzter Job, der kein Array ist und einige einfache Bereinigungsvorgänge durchführt und eine Amazon SNS SNS-Benachrichtigung mit einer Meldung sendet, dass die Pipeline abgeschlossen wurde, und einem Link zur Ausgabe-URL. Die SubmitJobJSON-Syntax lautet wie folgt.

```
{
    "jobName": "JobE",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobE-Cleanup-and-Notification:1",
    "parameters": {
        "SourceBucket": "s3://amzn-s3-demo-source-bucket",
        "Recipient": "pipeline-notifications@mycompany.com"
```

```
    },
    "dependsOn": [
        {
            "jobId": "JobD_job_ID"
        }
    ]
}
```

Tutorial: Verwenden Sie den Array-Jobindex, um die Jobdifferenzierung zu steuern

In diesem Tutorial wird beschrieben, wie Sie die `AWS_BATCH_JOB_ARRAY_INDEX` Umgebungsvariable verwenden, um die untergeordneten Jobs zu unterscheiden. Jeder untergeordnete Job ist dieser Variablen zugewiesen. In diesem Beispiel wird die Indexnummer des untergeordneten Jobs verwendet, um eine bestimmte Zeile in einer Datei zu lesen. Dann wird der mit dieser Zeilennummer verknüpfte Parameter durch einen Befehl im Container des Jobs ersetzt. Das Ergebnis ist, dass Sie mehrere AWS Batch Jobs haben können, die dasselbe Docker-Image und dieselben Befehlsargumente ausführen. Die Ergebnisse sind jedoch unterschiedlich, da der Array-Jobindex als Modifikator verwendet wird.

In diesem Tutorial erstellen Sie eine Textdatei, die – jeweils in einer eigenen Zeile stehend – alle Farben des Regenbogens umfasst. Anschließend erstellen Sie ein Einstiegsskript für einen Docker-Container, das den Index in einen Wert konvertiert, der für eine Zeilennummer in der Farbdatei verwendet werden kann. Der Index beginnt bei Null, aber die Zeilennummern beginnen bei Eins. Erstellen Sie ein Dockerfile, das die Farb- und Indexdateien in das Container-Image kopiert und `ENTRYPOINT` für das Bild das Entrypoint-Skript festlegt. Die Docker-Datei und die Ressourcen werden in einem Docker-Image erstellt, das an Amazon ECR übertragen wird. Anschließend registrieren Sie eine Jobdefinition, die Ihr neues Container-Image verwendet, reichen einen AWS Batch Array-Job mit dieser Jobdefinition ein und sehen sich die Ergebnisse an.

Themen

- Voraussetzungen
- Tutorial: Erstellen Sie ein Container-Image
- Tutorial: Übertragen Sie Ihr Bild auf Amazon ECR
- Tutorial: Eine Jobdefinition erstellen und registrieren
- Tutorial: Einen AWS Batch Array-Job einreichen

- Tutorial: Sehen Sie sich Ihre Array-Job-Logs an

Voraussetzungen

Für diesen Tutorial-Workflow gelten die folgenden Voraussetzungen:

- Eine AWS Batch Rechenumgebung. Weitere Informationen finden Sie unter Erstellen Sie eine Rechenumgebung.
- Eine AWS Batch Jobwarteschlange und eine zugehörige Rechenumgebung. Weitere Informationen finden Sie unter Eine Job-Warteschlange erstellen.
- Die AWS CLI ist auf Ihrem lokalen System installiert. Weitere Informationen finden Sie unter Installieren der AWS Command Line Interface im AWS Command Line Interface - Benutzerhandbuch.
- Das auf dem lokalen System installierte Docker. Weitere Informationen finden Sie unter Über Docker CE in der Docker-Dokumentation.

Tutorial: Erstellen Sie ein Container-Image

Sie können das `AWS_BATCH_JOB_ARRAY_INDEX` in einer Jobdefinition im Befehlsparameter verwenden. Wir empfehlen jedoch, stattdessen ein Container-Image zu erstellen, das die Variable in einem Entrypoint-Skript verwendet. In diesem Abschnitt wird beschrieben, wie Sie ein solches Container-Image erstellen.

So erstellen Sie ein Docker-Container-Image:

1. Erstellen Sie ein neues Verzeichnis zur Verwendung als Docker-Image-Workspace und wechseln Sie dort hin.
2. Erstellen Sie eine Datei mit dem Namen `colors.txt` in Ihrem Workspace-Verzeichnis und fügen Sie Folgendes ein.

```
red
orange
yellow
green
blue
indigo
violet
```

3. Erstellen Sie eine Datei mit dem Namen `print-color.sh` in Ihrem Workspace-Verzeichnis und fügen Sie Folgendes ein.

> **Note**
>
> Für die `LINE`-Variable ist `AWS_BATCH_JOB_ARRAY_INDEX` + 1 festgelegt, da der Array-Index bei 0 beginnt, die Zeilennummern aber bei 1. Die `COLOR` Variable wird auf die Farbe gesetzt`colors.txt`, die ihrer Zeilennummer zugeordnet ist.

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Erstellen Sie eine Datei mit dem Namen `Dockerfile` in Ihrem Workspace-Verzeichnis und fügen Sie den folgenden Inhalt ein. Diese Docker-Datei kopiert die vorherigen Dateien auf Ihren Container und legt fest, dass das entrypoint-Skript ausgeführt wird, wenn der Container gestartet wird.

```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Erstellen Sie das Docker-Image.

```
$ docker build -t print-color .
```

6. Testen Sie den Container mit dem folgenden Skript. Dieses Skript setzt die `AWS_BATCH_JOB_ARRAY_INDEX` Variable lokal auf 0 und erhöht sie dann, um zu simulieren, was ein Array-Job mit sieben untergeordneten Objekten bewirkt.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
    docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
    AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
```

```
done
```

Im Folgenden wird die Ausgabe dargestellt.

```
My favorite color of the rainbow is red.
My favorite color of the rainbow is orange.
My favorite color of the rainbow is yellow.
My favorite color of the rainbow is green.
My favorite color of the rainbow is blue.
My favorite color of the rainbow is indigo.
My favorite color of the rainbow is violet.
```

Tutorial: Übertragen Sie Ihr Bild auf Amazon ECR

Nachdem Sie Ihren Docker-Container erstellt und getestet haben, können Sie ihn in ein Image-Repository verschieben. In diesem Beispiel wird Amazon ECR verwendet, Sie können jedoch auch eine andere Registrierung verwenden, z. B. DockerHub

1. Erstellen Sie ein Amazon ECR-Image-Repository, um Ihr Container-Image zu speichern. In diesem Beispiel wird nur der verwendet AWS CLI, Sie können aber auch den AWS Management Console verwenden. Weitere Informationen finden Sie unter Creating a Repository im Amazon Elastic Container Registry-Benutzerhandbuch.

   ```
   $ aws ecr create-repository --repository-name print-color
   ```

2. Kennzeichnen Sie Ihr `print-color` Bild mit Ihrer Amazon ECR-Repository-URI, die im vorherigen Schritt zurückgegeben wurde.

   ```
   $ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
   ```

3. Melden Sie sich bei Ihrer Amazon ECR-Registrierung an. Weitere Informationen finden Sie unter Registry-Authentifizierung im Benutzerhandbuch zu Amazon-Elastic-Container-Registry.

   ```
   $ aws ecr get-login-password \
       --region region | docker login \
       --username AWS \
       --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
   ```

4. Pushen Sie Ihr Bild auf Amazon ECR.

```
$ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

Tutorial: Eine Jobdefinition erstellen und registrieren

Da sich Ihr Docker-Image nun in einer Image-Registry befindet, können Sie es in einer AWS Batch Jobdefinition angeben. Anschließend können Sie es später verwenden, um einen Array-Job auszuführen. In diesem Beispiel wird nur der verwendet AWS CLI. Sie können jedoch auch die verwenden AWS Management Console. Weitere Informationen finden Sie unter Erstellen Sie eine Auftragsdefinition mit einem Knoten .

So erstellen Sie eine Auftragsdefinition:

1. Erstellen Sie eine Datei mit dem Namen `print-color-job-def.json` in Ihrem Workspace-Verzeichnis und fügen Sie Folgendes ein. Ersetzen Sie den Image-Repository-URI durch den URI des Bildes.

```
{
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
    "resourceRequirements": [
        {
            "type": "MEMORY",
            "value": "250"
        },
        {
            "type": "VCPU",
            "value": "1"
        }
    ]
  }
}
```

2. Registrieren Sie die Jobdefinition bei AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-
def.json
```

Tutorial: Einen AWS Batch Array-Job einreichen

Nachdem Sie Ihre Jobdefinition registriert haben, können Sie einen AWS Batch Array-Job einreichen, der Ihr neues Container-Image verwendet.

Um einen AWS Batch Array-Job einzureichen

1. Erstellen Sie eine Datei mit dem Namen `print-color-job.json` in Ihrem Workspace-Verzeichnis und fügen Sie Folgendes ein.

   > **Note**
   >
   > In diesem Beispiel wird die im the section called “Voraussetzungen” Abschnitt erwähnte Job-Warteschlange verwendet.

   ```
   {
     "jobName": "print-color",
     "jobQueue": "existing-job-queue",
     "arrayProperties": {
       "size": 7
     },
     "jobDefinition": "print-color"
   }
   ```

2. Reichen Sie den Job in Ihre AWS Batch Job-Warteschlange ein. Notieren Sie sich die Job-ID, die in der Ausgabe zurückgegeben wird.

   ```
   $ aws batch submit-job --cli-input-json file://print-color-job.json
   ```

3. Beschreiben Sie den Status des Auftrags und warten Sie, bis der Auftrag den Status `SUCCEEDED` erhält.

Tutorial: Sehen Sie sich Ihre Array-Job-Logs an

Nachdem Ihr Job den `SUCCEEDED` Status erreicht hat, können Sie die CloudWatch Protokolle im Container des Jobs einsehen.

Um die Logs Ihres Jobs in CloudWatch Logs einzusehen

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie im linken Navigationsbereich Aufträge aus.
3. Wählen Sie für Auftragswarteschlange eine Warteschlange aus.
4. Wählen Sie im Bereich Status die Option erfolgreich aus.
5. Zum Anzeigen aller untergeordneten Aufträge für den Array-Auftrag wählen Sie die Auftrags-ID aus, die im vorherigen Abschnitt zurückgegeben wurde.
6. Zum Anzeigen der Protokolle aus dem Container des Auftrags wählen Sie einen der untergeordneten Aufträge und dann die Option Protokolle anzeigen aus.

7. Zeigen Sie die Protokolle der anderen untergeordneten Protokolle an. Jeder Auftrag gibt eine andere Farbe des Regenbogens zurück.

Führen Sie GPU-Jobs aus

GPU-Jobs helfen Ihnen dabei, Jobs auszuführen, die die einer Instanz verwenden GPUs.

Die folgenden EC2 GPU-basierten Amazon-Instance-Typen werden unterstützt. Weitere Informationen finden Sie unter Amazon EC2 G3-Instances, Amazon EC2 G4-Instances, Amazon EC2 G5-Instances, Amazon EC2 G6-Instances, Amazon P2-Instances, Amazon EC2 P3-Instances, Amazon EC2 EC2 P4d-Instances, Amazon EC2 P5-Instances, Amazon EC2 P6-Instances, Amazon Trn1-Instances, Amazon Trn2-Instances, Amazon Trn2-Instances, Amazon EC2 EC2 EC2 Inf1-Instances, Amazon EC2 Inf2-Instances, Amazon EC2 Dl1-Instances und Amazon EC2 Dl2-Instances.

Instance-Typ	GPUs	GPU-Arbeitsspeicher	v CPUs	Arbeitsspeicher	Netzwerkbandbreite
g3s.xgroß	1	8 GiB	4	30,5 GiB	10 Gbit/s
g3.4xgroß	1	8 GiB	16	122 GiB	Bis zu 10 Gbit/s
g3.8xgroß	2	16 GiB	32	244 GiB	10 Gbit/s
g3.16xgroß	4	32 GiB	64	488 GiB	25 Gbit/s
g4dn.xgroß	1	16 GiB	4	16 GiB	Bis zu 25 Gbit/s
g4dn.2xgroß	1	16 GiB	8	32 GiB	Bis zu 25 Gbit/s
g4dn.4xgroß	1	16 GiB	16	64 GiB	Bis zu 25 Gbit/s
g4dn.8xgroß	1	16 GiB	32	128 GiB	50 Gbit/s
g4dn.12xgroß	4	64 GiB	48	192 GiB	50 Gbit/s
g4dn.16xgroß	1	16 GiB	64	256 GiB	50 Gbit/s
g5.xgroß	1	24 GiB	4	16 GiB	Bis zu 10 Gbit/s
g5.2xlarge	1	24 GiB	8	32 GiB	Bis zu 10 Gbit/s
g5.4xlarge	1	24 GiB	16	64 GiB	Bis zu 25 Gbit/s
g5.8xlarge	1	24 GiB	32	128 GiB	25 Gbit/s
g5.16xlarge	1	24 GiB	64	256 GiB	25 Gbit/s
g5.12xlarge	4	96 GiB	48	192 GiB	40 GBit/s
g5.24xlarge	4	96 GiB	96	384 GiB	50 Gbit/s
g5.48xlarge	8	192 GiB	192	768 GiB	100 Gbit/s
g5g.xlarge	1	16 GB	4	8 GB	Bis zu 10 Gbit/s
g5g.2xlarge	1	16 GB	8	16 GB	Bis zu 10 Gbit/s

Instance-Typ	GPUs	GPU-Arbeitsspeicher	v CPUs	Arbeitsspeicher	Netzwerkbandbreite
g5g.4xlarge	1	16 GB	16	32 GB	Bis zu 10 Gbit/s
g5g.8xlarge	1	16 GB	32	64 GB	12 Gbit/s
g5g.16xlarge	2	32 GB	64	128 GB	25 Gbit/s
g5g.metal	2	32 GB	64	128 GB	25 Gbit/s
g6.x groß	1	24 GiB	4	16 GiB	Bis zu 10 Gbit/s
g 6.2 x groß	1	24 GiB	8	32 GiB	Bis zu 10 Gbit/s
g 6,4 x groß	1	24 GiB	16	64 GiB	Bis zu 25 Gbit/s
g 6,8 x groß	1	24 GiB	32	128 GiB	25 Gbit/s
g 6.16 x groß	1	24 GiB	64	256 GiB	25 Gbit/s
g 6.12 x groß	4	96 GiB	48	192 GiB	40 GBit/s
g 6,24 x groß	4	96 GiB	96	384 GiB	50 Gbit/s
g 6,48 x groß	8	192 GiB	192	768 GiB	100 Gbit/s
g6e.x groß	1	48 GB	4	32 GB	Bis zu 20 Gbit/s
g6e.2 x groß	1	48 GB	8	64 GB	Bis zu 20 Gbit/s
g6e.4x groß	1	48 GB	16	128 GB	20 Gbit/s
g6e.8 x groß	1	48 GB	32	256 GB	25 Gbit/s
G6E. 16 x groß	1	48 GB	64	512 GB	35 Gbit/s
g6e.12x groß	4	192 GB	48	384 GB	100 Gbit/s
g6e.24x groß	4	192 GB	96	768 GB	200 Gbit/s
g6e.48x groß	8	384 GB	192	1536 GB	400 Gbit/s

Instance-Typ	GPUs	GPU-Arbeitsspeicher	v CPUs	Arbeitsspeicher	Netzwerkbandbreite
gr 6,4 x groß	1	24 GB	16	128 GB	Bis zu 25 Gbit/s
gr 6,8 x groß	1	24 GB	32	256 GB	25 Gbit/s
p2.xgroß	1	12 GiB	4	61 GiB	Hoch
p2.8xgroß	8	96 GiB	32	488 GiB	10 Gbit/s
p2.16xgroß	16	192 GiB	64	732 GiB	20 Gbit/s
p3.2xgroß	1	16 GiB	8	61 GiB	Bis zu 10 Gbit/s
p3.8xgroß	4	64 GiB	32	244 GiB	10 Gbit/s
p3.16xgroß	8	128 GiB	64	488 GiB	25 Gbit/s
p3dn.24xgroß	8	256 GiB	96	768 GiB	100 Gbit/s
p4d.24xgroß	8	320 GB	96	152 GB	400 Gbit/s
p4de.24xlarge	8	640 GB	96	152 GB	400 Gbit/s
p5.48xlarge	8	640 GiB	192	2 TiB	3200 Gbit/s
p 5e. 48 x groß	8	128 GB	192	2 TB	3200 Gbit/s
p 5 en 48 x groß	8	128 GB	192	2 TB	3200 Gbit/s
p6-b 200.48x groß	8	140 GB	192	2 TB	100 Gbit/s
trn1.2xlarge	1	32 GB	8	32 GB	Bis zu 12,5 GBit/s
trn1.32xlarge	16	512 GB	128	512 GB	800 Gbit/s
trn1n.32xlarge	16	512 GB	128	512 GB	1600 Gbit/s
TRN 2,48 x groß	16	1.5 TB	192	2 TB	3,2 Tbit/s

Instance-Typ	GPUs	GPU-Arbeitsspeicher	v CPUs	Arbeitsspeicher	Netzwerkbandbreite
inf1.xlarge	1	8 GB	4	8 GB	Bis zu 25 Gbit/s
inf1.2xlarge	1	8 GB	8	16 GB	Bis zu 25 Gbit/s
inf1.6xlarge	4	32 GB	24	48 GB	25 Gbit/s
inf1.24xlarge	16	128 GB	96	192 GB	100 Gbit/s
inf2.xlarge	1	32 GB	4	16 GB	Bis zu 15 GBit/s
inf2.8xlarge	1	32 GB	32	128 GB	Bis zu 25 Gbit/s
inf2.24xlarge	6	192 GB	96	384 GB	50 Gbit/s
inf2.48xlarge	12	384 GB	192	768 GB	100 Gbit/s
dl1.24xlarge	8	256 GB	96	768 GB	400 Gbit/s
dl2q.24xlarge	8	128 GB	96	768 GB	100 Gbit/s

Note

Für GPU-Jobs werden AWS Batch nur Instance-Typen unterstützt, die über NVIDIA verfügen GPUs. Die `G4ad`Familie wird beispielsweise nicht für die GPU-Planung unterstützt. Sie können weiterhin `G4ad`on verwenden, AWS Batch indem Sie nur die vCPU- und Speicheranforderungen in der Jobdefinition definieren und dann GPUs direkt auf den Host zugreifen, indem Sie Benutzerdaten in einer EC2 Amazon-Startvorlage mit einem für die Datenverarbeitung optimierten Amazon ECS- oder Amazon EKS-AMI oder einem benutzerdefinierten AMI für die Verwendung von AMD GPUs anpassen.
Instance-Typen, die eine ARM64 Architektur verwenden, werden für GPU-Jobs mit kundenspezifischen AMIs Benutzerdaten AWS Batch oder EC2 Amazon-Benutzerdaten unterstützt, auf die GPUs durch benutzerdefinierten Code und Konfigurationen zugegriffen werden kann. Zum Beispiel die `G5g`Instance-Familie.

Der ResourceRequirements-Parameter für die Jobdefinition gibt die Anzahl der an den Container GPUs anzuheften an. Diese Anzahl von GPUs ist für die Dauer dieses Auftrags für keinen anderen Job verfügbar, der auf dieser Instance ausgeführt wird. Alle Instance-Typen in einer Rechenumgebung, die GPU-Jobs ausführen`p3`, müssen zu den `g6` Instance-Familien `p4` `p5` `p6g3`,`g3s`,`g4`,`g5`,,, oder gehören. Geschieht dies nicht, kann es sein, dass ein GPU-Job im `RUNNABLE` Status hängen bleibt.

Jobs, die das nicht verwenden, GPUs können auf GPU-Instanzen ausgeführt werden. Ihre Ausführung auf GPU-Instanzen kann jedoch mehr kosten als auf ähnlichen Nicht-GPU-Instanzen. Abhängig von der spezifischen vCPU, dem Arbeitsspeicher und der benötigten Zeit können diese Nicht-GPU-Jobs die Ausführung von GPU-Jobs blockieren.

Themen

- Tutorial: Einen GPU-basierten Kubernetes Cluster auf Amazon EKS erstellen
- Tutorial: Erstellen Sie eine Amazon EKS-GPU-Auftragsdefinition
- Tutorial: Führen Sie einen GPU-Job in Ihrem Amazon EKS-Cluster aus

Tutorial: Einen GPU-basierten Kubernetes Cluster auf Amazon EKS erstellen

Bevor Sie einen GPU-basierten Kubernetes Cluster auf Amazon EKS erstellen, müssen Sie die Schritte unter abgeschlossen haben. Erste Schritte mit AWS Batch Amazon EKS Beachten Sie außerdem Folgendes:

- AWS Batch unterstützt Instance-Typen mit NVIDIA GPUs.
- AWS Batch Wählt standardmäßig das Amazon EKS-beschleunigte AMI mit der Kubernetes Version aus, die Ihrer Amazon EKS-Cluster-Steuerebenenversion entspricht.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
```

```
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 1024,
    "instanceTypes": [
      "p3dn.24xlarge",
      "p4d.24xlarge"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF

$ aws batch create-compute-environment --cli-input-json file://./batch-eks-gpu-ce.json
```

AWS Batch verwaltet das NVIDIA GPU-Geräte-Plugin nicht in Ihrem Namen. Sie müssen dieses Plugin in Ihrem Amazon EKS-Cluster installieren und zulassen, dass es auf die AWS Batch Knoten abzielt. Weitere Informationen finden Sie unter Enabling GPU Support in Kubernetes on GitHub.

Führen Sie die folgenden Befehle aus, um NVIDIA das Geräte-Plugin (`DaemonSet`) so zu konfigurieren, dass es auf die AWS Batch Knoten abzielt.

```
# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-
device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml
```

Es wird nicht empfohlen, rechenbasierte Workloads (CPU und Arbeitsspeicher) mit GPU-basierten Workloads in derselben Kombination aus Rechenumgebung und Jobwarteschlange zu kombinieren. Das liegt daran, dass Rechenaufträge GPU-Kapazität verbrauchen können.

Führen Sie die folgenden Befehle aus, um Jobwarteschlangen anzuhängen.

```
$ cat <<EOF > ./batch-eks-gpu-jq.json
 {
    "jobQueueName": "My-Eks-GPU-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-GPU-CE1"
      }
    ]
  }
EOF

$ aws batch create-job-queue --cli-input-json file://./batch-eks-gpu-jq.json
```

Tutorial: Erstellen Sie eine Amazon EKS-GPU-Auftragsdefinition

Derzeit `nvidia.com/gpu` wird nur unterstützt, und der von Ihnen festgelegte Ressourcenwert muss eine ganze Zahl sein. Sie können keine Bruchteile der GPU verwenden. Weitere Informationen finden Sie GPUs in der KubernetesDokumentation unter Zeitplan.

Führen Sie die folgenden Befehle aus, um eine GPU-Auftragsdefinition für Amazon EKS zu registrieren.

```
$ cat <<EOF > ./batch-eks-gpu-jd.json
{
    "jobDefinitionName": "MyGPUJobOnEks_Smi",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "hostNetwork": true,
            "containers": [
                {
                    "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
                    "command": ["nvidia-smi"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi",
                            "nvidia.com/gpu": "1"
```

```
                        }
                    }
                }
            ]
        }
    }
}
EOF

$ aws batch register-job-definition --cli-input-json file://./batch-eks-gpu-jd.json
```

Tutorial: Führen Sie einen GPU-Job in Ihrem Amazon EKS-Cluster aus

Die GPU-Ressource ist nicht komprimierbar. AWS Batch erstellt eine Pod-Spezifikation für GPU-Jobs, bei der der Wert der Anfrage dem Wert der Grenzwerte entspricht. Dies ist eine Kubernetes Anforderung.

Führen Sie die folgenden Befehle aus, um einen GPU-Job einzureichen.

```
$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJobOnEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
 with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
 {podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}
```

Suchen Sie nach AWS Batch Aufträgen in einer Jobwarteschlange

AWS Batch Mit der Jobsuche können Sie Ihre Job suchen und filtern. Diese Funktion bietet die Möglichkeit, in einer bereits vorhandenen Auftragswarteschlange zu suchen und die zugehörigen Jobs nach einer von fünf Optionen zu filtern (Erstellt nach, Erstellt vor, Jobname, Jobdefinition und Status). Damit der Prozess diese Funktion nutzen kann, müssen Sie den Tutorial: AWS Batch Jobs suchen und filtern Vorgang abschließen.

Verwenden Sie die Beta-Job-Suchfunktion, um nach mehreren Kriterien gleichzeitig zu suchen. Sie können beispielsweise einen oder alle der folgenden Filter einbeziehen: Jobstatus, Datumsbereich

und zusätzliche Kriterien (wie Jobname, Jobdefinition oder Job-ID). Informationen zur Verwendung dieser Funktion finden Sie im Tutorial: (Beta) AWS Batch Jobs suchen und filtern Verfahren.

Themen

- Tutorial: AWS Batch Jobs suchen und filtern
- Tutorial: (Beta) AWS Batch Jobs suchen und filtern

Tutorial: AWS Batch Jobs suchen und filtern

Gehen Sie wie folgt vor, um alle Jobs in einer Job-Warteschlange in der AWS Batch Konsole aufzulisten. Verwenden Sie optional die Dropdownliste Filtertyp, um die Ergebnisse anhand der von Ihnen angegebenen Kriterien einzugrenzen.

1. Navigieren Sie zur AWS Batch -Konsole.
2. Wählen Sie im Navigationsbereich Jobs aus.
3. Erweitern Sie die Dropdownliste Job-Warteschlange und wählen Sie die Job-Warteschlange aus, in der Sie suchen möchten.

   > **Note**
   >
   > Sie können jeweils nur in einer Job-Warteschlange nach Jobs suchen.

4. (Optional) Wählen Sie den Filtertyp aus, den Sie anwenden möchten.

   - Erstellt nach: Gibt Jobs zurück, die nach einem bestimmten Datum und einer bestimmten Uhrzeit (in UTC) erstellt wurden. Wenn Sie diese Option auswählen, werden zwei Felder mit Filterwerten angezeigt, in denen Sie das früheste Datum und die Uhrzeit angeben können, zu der die Suche zurückgegeben wird.
   - Vorher erstellt: Gibt Jobs zurück, die vor einem bestimmten Datum und einer bestimmten Uhrzeit (in UTC) erstellt wurden. Wenn Sie diese Option auswählen, werden zwei Felder mit Filterwerten angezeigt, in denen Sie das späteste Datum und die Uhrzeit angeben können, zu der die Suche zurückgegeben werden soll.
   - Jobdefinition: Gibt Jobs zurück, die mit einem bestimmten Jobdefinitionsnamen erstellt wurden. Wenn Sie diese Option auswählen, wird ein Feld angezeigt, in dem Sie die vollständige oder teilweise Definition eines Jobs angeben können (wenn Sie am Ende ein Sternchen (*) hinzufügen, entspricht der Filter jeder Jobdefinition, die mit der Zeichenfolge vor dem Sternchen beginnt).

- Jobname: Gibt Jobs zurück, die einen bestimmten Jobnamen haben. Wenn Sie diese Option auswählen, wird ein Feld angezeigt, in dem Sie den vollständigen oder teilweisen Namen eines Jobs angeben können (wenn Sie am Ende ein Sternchen (*) hinzufügen, passt der Filter auf jeden Jobnamen, der mit der Zeichenfolge vor dem Sternchen beginnt).
- Status: Gibt Jobs zurück, die den ausgewählten Jobstatus haben. Wenn Sie diese Option auswählen, wird eine Dropdownliste angezeigt, die Sie erweitern und einen bestimmten Status auswählen können. Weitere Informationen finden Sie unter Auftragsstatus.

5. (Optional) Wählen Sie Alle Jobs laden, um alle Jobs in einer Job-Warteschlange der letzten sieben Tage ohne die vorherigen Filter zurückzugeben. Wenn Sie diese Option wählen, wird die folgende Meldung angezeigt:

> **Note**
>
> AWS Batch Speichert jetzt den Jobverlauf von sieben (7) Tagen. Wenn Sie sich dafür entscheiden, alle Jobs zu laden, verlängert sich die Zeit, die Sie benötigen, um auf Ihre Jobs zuzugreifen und mit der Suche und Filterung zu beginnen. Wenn Sie stattdessen einen Filtertyp verwenden, erhalten Sie unmittelbarere Ergebnisse. Möchten Sie wirklich alle Jobs laden?

A Wählen Sie Bestätigen, um die Nachricht zu schließen.

- Wenn Sie diese Option wählen, wird eine Dropdownliste angezeigt, in der Sie ähnliche Filteroptionen auswählen können: Status, Jobname und Job-ID. Wenn Sie eine dieser Optionen wählen, können Sie Ihre Suchkriterien weiter filtern, wobei der angegebene Filter den von Ihnen eingegebenen Kriterien entspricht, ihnen nicht entspricht, sie enthält oder nicht enthält.

6. Wenn die Optionen vollständig sind, wählen Sie Suchen aus.

Tutorial: (Beta) AWS Batch Jobs suchen und filtern

Gehen Sie wie folgt vor, um alle Jobs in einer Auftragswarteschlange in der AWS Batch Konsole aufzulisten und die Beta-Suchfunktionen zu verwenden. Verwenden Sie optional den Abschnitt Suchoptionen, um die Ergebnisse anhand der von Ihnen angegebenen Kriterien einzugrenzen. Mit dieser Beta-Suchfunktion können Sie Suchkriterien kombinieren, um die Ergebnisse einzugrenzen. Sie können jedoch nur ein Kriterium pro Filter definieren.

1. Navigieren Sie zur AWS Batch -Konsole.
2. Wählen Sie im Navigationsbereich Jobs aus.
3. Stellen Sie sicher, dass die Option Jobs suchen — Beta aktiv ist.
4. Erweitern Sie die Dropdownliste Job-Warteschlange und wählen Sie die Job-Warteschlange aus, in der Sie suchen möchten.

> **Note**
>
> Sie können jeweils nur in einer Job-Warteschlange nach Jobs suchen.

5. (Optional) Erweitern Sie die Dropdownliste Jobstatus und wählen Sie einen oder mehrere Jobstatus aus. Weitere Informationen finden Sie unter Auftragsstatus.
6. (Optional) Wählen Sie Datumsbereich aus, um die Ergebnisse nach Datum und Uhrzeit zu filtern.
   - Wählen Sie den Modus Relativ, um nach Jobs zu suchen, die innerhalb eines Zeitraums ausgeführt wurden, der vom aktuellen Datum und der aktuellen Uhrzeit rückwärts gezählt wird.
   - Wählen Sie den Modus Absolut, um nach Jobs zu suchen, die innerhalb eines von Ihnen angegebenen Datums- und Zeitbereichs ausgeführt wurden.
7. (Optional) Geben Sie im Feld Zusätzliche Kriterien Stichwörter ein, die in die Suchergebnisse aufgenommen werden sollen. Sie können dieses Feld beispielsweise verwenden, um nach Jobname, Jobdefinition oder Job-ID zu suchen. Je nach Eigenschaft müssen Sie möglicherweise weitere Operatoren definieren, z. B. ist gleich (=) oder beginnt mit (^).
8. Wenn die Optionen vollständig sind, wählen Sie Suchen aus.

> **Tip**
>
> Sie können das Einstellungssymbol wählen, um die angezeigten Spalten und ihre Reihenfolge in der Tabelle mit den Suchergebnissen anzupassen.

Netzwerkmodi für AWS Batch Jobs

In der folgenden Tabelle werden die Netzwerkmodi und die typische Verwendung für AWS Batch Jobtypen beschrieben. Weitere Informationen zu Überlegungen und Verhaltensweisen finden Sie unter den Links in der Spalte „Jobtyp“.

Aufgabentyp	Unterstützte Netzwerkmodi	Typische Verwendung
ECS — EC2 einfacher Job	`host`	Wird für die am höchsten skalierbaren, peinlich parallel Batch-Workloads verwendet , die nur den Ausgang zu einer in Compute Environment definierten VPC erfordern.
ECS — EC2 Paralleljob mit mehreren Knoten	`awsvpc`	Wird für eng gekoppelt e, verteilte Workloads mit mehreren Hosts (Knoten) verwendet, die als einzelner Job mit koordinierter Kommunikation zwischen Task-Knoten modelliert werden.
ECS-Fargate einfache Aufgabe	`awsvpc`	Echt serverlos für peinlich parallel Batch-Workloads. Typischerweise das Jobmodell mit den niedrigsten Gesamtbet riebskosten und der höchsten Containerisolierung.
EKS — EC2 einfacher Job	Host und Pod	Wird für hoch skalierbare, peinlich parallel Batch-Wor kloads verwendet, die nur einen Ausgang zu einer in Compute Environment definierten VPC erfordern. Die Standardeinstellung ist Host-Netzwerk.
EC2 EKS-Paralleljob mit mehreren Knoten	Host und Pod	Wird für eng gekoppelt e, verteilte Workloads mit mehreren Hosts (Knoten)

Aufgabentyp	Unterstützte Netzwerkmodi	Typische Verwendung
		verwendet, die als ein einziger Job mit koordinierter Kommunikation zwischen Pod-Knoten modelliert werden. Die Standardeinstellung ist Host-Netzwerke.

Tutorial: AWS Batch Jobprotokolle an CloudWatch Logs senden

Sie können Ihre AWS Batch Jobs so konfigurieren, dass Protokollinformationen an Amazon CloudWatch Logs gesendet werden. Auf diese Weise können Sie verschiedene Protokolle Ihrer Jobs bequem an einem zentralen Ort einsehen. Weitere Informationen finden Sie unter Verwenden von CloudWatch Protokollen mit AWS Batch.

Sie können auch Job-Logs in der AWS Batch Konsole verwenden, um einen AWS Batch Job zu überwachen oder Fehler zu beheben.

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie Jobs (Aufträge) aus.
3. Wählen Sie Job Job-Warteschlange die gewünschte Job-Warteschlange aus.

   > **Tip**
   >
   > Wenn sich in der Auftragswarteschlange mehrere Jobs befinden, können Sie Suchen und Filtern aktivieren, um einen Job schneller zu finden. Weitere Informationen finden Sie unter Suchen Sie nach AWS Batch Aufträgen in einer Jobwarteschlange.

4. Wählen Sie unter Status den gewünschten Auftragsstatus aus.
5. Wählen Sie den gewünschten Job aus.
6. Scrollen Sie auf der Detailseite nach unten zu Job Logs.
7. Wählen Sie Protokolle abrufen aus.
8. Geben Sie für Autorisierung erforderlich ein und wählen Sie dann Autorisieren **OK**, um CloudWatch Amazon-Gebühren zu akzeptieren.

> **Note**
>
> Um Ihre Autorisierung für CloudWatch Gebühren zu widerrufen:
>
> 1. Wählen Sie im linken Navigationsbereich die Option Berechtigungen aus.
> 2. Wählen Sie für Job-Logs die Option Bearbeiten aus.
> 3. Deaktivieren Sie das CloudWatch Kontrollkästchen Batch zur Verwendung autorisieren.
> 4. Wählen Sie Änderungen speichern aus.

9. Überprüfen Sie die Protokolldaten für den AWS Batch Job.

> **Tip**
>
> Sie können das Protokoll nach Schlüsselwörtern, Max. Ergebnissen und Sortierung filtern. Sie können auch eines der Standardzeitintervalle wählen oder ein benutzerdefiniertes Intervall erstellen, um die Ergebnisse anzupassen.

Tutorial: AWS Batch Jobinformationen überprüfen

Sie können AWS Batch Jobinformationen wie Status, Jobdefinition und Containerinformationen überprüfen.

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie Jobs (Aufträge) aus.
3. Wählen Sie Job Job-Warteschlange die gewünschte Job-Warteschlange aus.

> **Tip**
>
> Wenn sich mehrere Jobs in der Auftragswarteschlange befinden, können Sie Suchen und Filtern aktivieren, um einen Job schneller zu finden. Weitere Informationen finden Sie unter Suchen Sie nach AWS Batch Aufträgen in einer Jobwarteschlange.

4. Wählen Sie den gewünschten Job aus.

> **Note**
>
> Sie können auch das AWS Command Line Interface (AWS CLI) verwenden, um Details zu einem AWS Batch Job anzuzeigen. Weitere Informationen finden Sie unter describe-jobs in der AWS CLI Befehlsreferenz.

Sicherheit in AWS Batch

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst 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 AWS Batch, finden Sie unter AWS Services im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud: Ihr Verantwortungsumfang wird durch den AWS -Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, einschließlich der 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 Verwendung anwenden können AWS Batch. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen AWS Batch , 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 AWS Batch Ressourcen unterstützen.

Themen

- Identity and Access Management für AWS Batch
- AWS Batch IAM-Richtlinien, -Rollen und -Berechtigungen
- AWS Batch IAM-Ausführungsrolle
- Erstellen einer Virtual Private Cloud
- Verwenden Sie einen Schnittstellenendpunkt für Access AWS Batch
- Konformitätsprüfung für AWS Batch
- Sicherheit der Infrastruktur in AWS Batch
- Serviceübergreifende Confused-Deputy-Prävention

- AWS Batch API-Aufrufe protokollieren mit AWS CloudTrail
- Problembehandlung bei AWS Batch IAM

Identity and Access Management für AWS Batch

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. AWS Batch IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Wie AWS Batch funktioniert mit IAM
- Beispiele für identitätsbasierte Richtlinien für AWS Batch
- AWS verwaltete Richtlinien für AWS Batch

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. AWS Batch

Dienstbenutzer — Wenn Sie den AWS Batch Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr AWS Batch Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Unter Problembehandlung bei AWS Batch IAM finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Feature in AWS Batch haben.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die AWS Batch Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf AWS Batch. Es ist Ihre Aufgabe, zu bestimmen, auf welche AWS Batch Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die

Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen kann AWS Batch, finden Sie unterWie AWS Batch funktioniert mit IAM.

IAM-Administrator: Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf AWS Batch verfassen können. Beispiele für AWS Batch identitätsbasierte Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Batch

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter AWS Signature Version 4 für API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center -Benutzerhandbuch und AWS Multi-Faktor-Authentifizierung (MFA) in IAM im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter Regelmäßiges

Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine IAM-Gruppe ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter Methoden für die Übernahme einer Rolle im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter Erstellen von Rollen für externe Identitätsanbieter (Verbund) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.

- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Service aufrufen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Forward Access Sessions (FAS) — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle – Eine Servicerolle ist eine IAM-Rolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.
  - Dienstbezogene Rolle — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein

Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. 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

Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter Auswählen zwischen verwalteten und eingebundenen Richtlinien im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im IAM-Benutzerhandbuch.
- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter Resource Control Policies (RCPs) im AWS Organizations Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und

der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter Bewertungslogik für Richtlinien.

Wie AWS Batch funktioniert mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren AWS Batch, mit welchen IAM-Funktionen Sie arbeiten können. AWS Batch

IAM-Funktionen, die Sie mit verwenden können AWS Batch

IAM-Feature	AWS Batch Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie AWS Batch und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für AWS Batch

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. 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 Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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 AWS Batch

Beispiele für AWS Batch identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Batch

Politische Maßnahmen für AWS Batch

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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 der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der AWS Batch Aktionen finden Sie unter Definierte Aktionen von AWS Batch in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix AWS Batch verwendet:

```
batch
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "batch:action1",
      "batch:action2"
         ]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "batch:Describe*"
```

Beispiele für AWS Batch identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS Batch

Politische Ressourcen für AWS Batch

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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": "*"
```

Eine Liste der AWS Batch Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Resources Defined by AWS Batch in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von AWS Batch definierte Aktionen.

Richtlinien-Bedingungsschlüssel für AWS Batch

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann 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, AWS wertet die Bedingung mithilfe 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. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter IAM-Richtlinienelemente: Variablen und Tags im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Eine Liste der AWS Batch Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS Batch in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Definierte Aktionen von AWS Batch.

Attributbasierte Zugriffskontrolle (ABAC) mit AWS Batch

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

Verwenden Sie temporäre Anmeldeinformationen mit AWS Batch

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln von einer Benutzerrolle zu einer IAM-Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipal-Berechtigungen für AWS Batch

Unterstützt Forward Access Sessions (FAS): Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AWS Batch

Unterstützt Servicerollen: Ja

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. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die AWS Batch Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn AWS Batch eine Anleitung dazu gibt.

Mit Diensten verknüpfte Rollen für AWS Batch

Unterstützt dienstbezogene Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS Batch

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, AWS Batch -Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden AWS Batch, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Batch in der Service Authorization Reference.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Konsole AWS Batch
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS Batch Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue

und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.

- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Konsole AWS Batch

Um auf die AWS Batch Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS Batch Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die AWS Batch Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die AWS Batch `ConsoleAccess` oder die `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI AWS OR-API.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS verwaltete Richtlinien für AWS Batch

Sie können AWS verwaltete Richtlinien verwenden, um die Verwaltung des Identitätszugriffs für Ihr Team und die bereitgestellten AWS Ressourcen zu vereinfachen. AWS verwaltete Richtlinien decken eine Vielzahl gängiger Anwendungsfälle ab, sind standardmäßig in Ihrem AWS Konto verfügbar und werden in Ihrem Namen verwaltet und aktualisiert. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Wenn Sie mehr Flexibilität benötigen, können Sie sich alternativ dafür entscheiden, vom Kunden verwaltete IAM-Richtlinien zu erstellen. Auf diese Weise können

Sie Ihrem Team bereitgestellte Ressourcen nur mit genau den Berechtigungen ausstatten, die es benötigt.

Weitere Informationen zu AWS verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS Services verwalten und aktualisieren AWS verwaltete Richtlinien in Ihrem Namen. In regelmäßigen Abständen fügen AWS Dienste einer AWS verwalteten Richtlinie zusätzliche Berechtigungen hinzu. AWS verwaltete Richtlinien werden höchstwahrscheinlich aktualisiert, wenn eine neue Funktion gestartet oder ein neuer Vorgang verfügbar wird. Diese Updates wirken sich automatisch auf alle Identitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. Sie entfernen jedoch weder Berechtigungen noch beeinträchtigen sie Ihre bestehenden Berechtigungen.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die `ReadOnlyAccess` AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS -Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

AWS verwaltete Richtlinie: BatchServiceRolePolicy

Die BatchServiceRolePolicyverwaltete IAM-Richtlinie wird von der `AWSServiceRoleForBatch`serviceverknüpften Rolle verwendet. Auf diese Weise können AWS Batch Sie Aktionen in Ihrem Namen ausführen. Sie können diese Richtlinie nicht mit Ihren IAM-Entitäten verknüpfen. Weitere Informationen finden Sie unter Verwenden Sie dienstbezogene Rollen für AWS Batch.

Diese Richtlinie ermöglicht AWS Batch es, die folgenden Aktionen an bestimmten Ressourcen durchzuführen:

- `autoscaling`— Ermöglicht das AWS Batch Erstellen und Verwalten von Amazon EC2 Auto Scaling Scaling-Ressourcen. AWS Batch erstellt und verwaltet Amazon EC2 Auto Scaling Scaling-Gruppen für die meisten Computerumgebungen.

- `ec2`— Ermöglicht AWS Batch die Kontrolle des Lebenszyklus von EC2 Amazon-Instances sowie die Erstellung und Verwaltung von Startvorlagen und Tags. AWS Batch erstellt und verwaltet EC2 Spot-Flottenanfragen für einige EC2 Spot-Computerumgebungen.
- `ecs`- Ermöglicht AWS Batch die Erstellung und Verwaltung von Amazon ECS-Clustern, Aufgabendefinitionen und Aufgaben für die Auftragsausführung.
- `eks`- Ermöglicht AWS Batch die Beschreibung der Amazon EKS-Cluster-Ressource für Validierungen.
- `iam`- Ermöglicht AWS Batch die Validierung und Weitergabe von Rollen, die vom Eigentümer bereitgestellt wurden, an Amazon EC2, Amazon EC2 Auto Scaling und Amazon ECS.
- `logs`— Ermöglicht AWS Batch das Erstellen und Verwalten von Protokollgruppen und Protokollströmen für AWS Batch Jobs.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSBatchPolicyStatement1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSpotFleetRequestHistory",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:RequestSpotFleet",
                "autoscaling:DescribeAccountLimits",
```

```
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "eks:DescribeCluster",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeTasks",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:ListTaskDefinitionFamilies",
                "ecs:ListTaskDefinitions",
                "ecs:ListTasks",
                "ecs:DeregisterTaskDefinition",
                "ecs:TagResource",
                "ecs:ListAccountSettings",
                "logs:DescribeLogGroups",
                "iam:GetInstanceProfile",
                "iam:GetRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSBatchPolicyStatement2",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*"
        },
        {
            "Sid": "AWSBatchPolicyStatement3",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*:log-stream:*"
        },
        {
            "Sid": "AWSBatchPolicyStatement4",
            "Effect": "Allow",
            "Action": [
```

```
                "autoscaling:CreateOrUpdateTags"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement5",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
                        "ecs-tasks.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement6",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "autoscaling.amazonaws.com",
                        "ecs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement7",
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateLaunchTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement8",
            "Effect": "Allow",
            "Action": [
                "ec2:TerminateInstances",
                "ec2:CancelSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "ec2:DeleteLaunchTemplate"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement9",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteLaunchConfiguration"
            ],
            "Resource":
 "arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/
AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement10",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
```

```
                "autoscaling:SetDesiredCapacity",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:SuspendProcesses",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:TerminateInstanceInAutoScalingGroup"
            ],
            "Resource":
 "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement11",
            "Effect": "Allow",
            "Action": [
                "ecs:DeleteCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:cluster/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement12",
            "Effect": "Allow",
            "Action": [
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:task-definition/*"
        },
        {
            "Sid": "AWSBatchPolicyStatement13",
            "Effect": "Allow",
            "Action": [
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:task/*/*"
        },
        {
            "Sid": "AWSBatchPolicyStatement14",
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
```

```
                "ecs:RegisterTaskDefinition"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement15",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:launch-template/*",
                "arn:aws:ec2:*:*:placement-group/*",
                "arn:aws:ec2:*:*:capacity-reservation/*",
                "arn:aws:ec2:*:*:elastic-gpu/*",
                "arn:aws:elastic-inference:*:*:elastic-inference-accelerator/*",
                "arn:aws:resource-groups:*:*:group/*"
            ]
        },
        {
            "Sid": "AWSBatchPolicyStatement16",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement17",
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "RunInstances",
                        "CreateLaunchTemplate",
                        "RequestSpotFleet"
                    ]
                }
            }
        }
    ]
}
```

AWS verwaltete Richtlinie: AWSBatchServiceRoleRichtlinie

Die genannte Richtlinie für Rollenberechtigungen AWSBatchServiceRole AWS Batch ermöglicht es, die folgenden Aktionen für bestimmte Ressourcen durchzuführen:

Die AWSBatchServiceRoleverwaltete IAM-Richtlinie wird häufig von einer Rolle mit dem Namen verwendet AWSBatchServiceRoleund umfasst die folgenden Berechtigungen. Nach den standardmäßigen Sicherheitshinweisen zur Erteilung von geringsten Privilegien kann die von AWSBatchServiceRole verwaltete Richtlinie als Leitfaden verwendet werden. Wenn eine der Berechtigungen, die in der verwalteten Richtlinie erteilt werden, für Ihren Anwendungsfall nicht erforderlich ist, erstellen Sie eine benutzerdefinierte Richtlinie, und fügen Sie nur die erforderlichen Berechtigungen hinzu. Diese AWS Batch verwaltete Richtlinie und Rolle können für die meisten Arten von Computerumgebungen verwendet werden, aber die Verwendung von servicebasierten Rollen wird bevorzugt, um eine weniger fehleranfällige, umfassendere und besser verwaltete Erfahrung zu erzielen.

- `autoscaling`— Ermöglicht das AWS Batch Erstellen und Verwalten von Amazon EC2 Auto Scaling Scaling-Ressourcen. AWS Batch erstellt und verwaltet Amazon EC2 Auto Scaling Scaling-Gruppen für die meisten Computerumgebungen.

- `ec2`— Ermöglicht AWS Batch die Verwaltung des Lebenszyklus von EC2 Amazon-Instances sowie die Erstellung und Verwaltung von Startvorlagen und Tags. AWS Batch erstellt und verwaltet EC2 Spot-Flottenanfragen für einige EC2 Spot-Computerumgebungen.
- `ecs`- Ermöglicht AWS Batch die Erstellung und Verwaltung von Amazon ECS-Clustern, Aufgabendefinitionen und Aufgaben für die Auftragsausführung.
- `iam`- Ermöglicht AWS Batch die Validierung und Weitergabe von Rollen, die vom Eigentümer bereitgestellt wurden, an Amazon EC2, Amazon EC2 Auto Scaling und Amazon ECS.
- `logs`— Ermöglicht AWS Batch das Erstellen und Verwalten von Protokollgruppen und Protokollströmen für AWS Batch Jobs.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSBatchPolicyStatement1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSpotFleetRequestHistory",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:CreateLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:RequestSpotFleet",
                "ec2:CancelSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
```

```
                "ec2:TerminateInstances",
                "ec2:RunInstances",
                "autoscaling:DescribeAccountLimits",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:SuspendProcesses",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeTasks",
                "ecs:ListAccountSettings",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:ListTaskDefinitionFamilies",
                "ecs:ListTaskDefinitions",
                "ecs:ListTasks",
                "ecs:CreateCluster",
                "ecs:DeleteCluster",
                "ecs:RegisterTaskDefinition",
                "ecs:DeregisterTaskDefinition",
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:UpdateContainerAgent",
                "ecs:DeregisterContainerInstance",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
                "iam:GetInstanceProfile",
                "iam:GetRole"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "AWSBatchPolicyStatement2",
            "Effect": "Allow",
            "Action": "ecs:TagResource",
            "Resource": [
                "arn:aws:ecs:*:*:task/*_Batch_*"
            ]
        },
        {
            "Sid": "AWSBatchPolicyStatement3",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
                        "ecs-tasks.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement4",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "autoscaling.amazonaws.com",
                        "ecs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement5",
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "RunInstances"
                }
            }
        }
    ]
}
```

AWS verwaltete Richtlinie: AWSBatchFullAccess

Die AWSBatchFullAccessRichtlinie gewährt AWS Batch Aktionen uneingeschränkten Zugriff auf AWS Batch Ressourcen. Es gewährt auch Zugriff auf Beschreiben und Auflisten von Aktionen für Amazon-EC2, Amazon ECS-, Amazon EKS- und IAM-Services. CloudWatch Auf diese Weise können IAM-Identitäten, entweder Benutzer oder Rollen, AWS Batch verwaltete Ressourcen einsehen, die in ihrem Namen erstellt wurden. Schließlich ermöglicht diese Richtlinie auch, dass ausgewählte IAM-Rollen an diese Dienste übergeben werden.

Sie können eine Verbindung AWSBatchFullAccesszu Ihren IAM-Entitäten herstellen. AWS Batch ordnet diese Richtlinie auch einer Servicerolle zu, mit der Sie Aktionen AWS Batch in Ihrem Namen ausführen können.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
```

```
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource":"*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":[
        "arn:aws:iam::*:role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/ecsInstanceRole",
        "arn:aws:iam::*:instance-profile/ecsInstanceRole",
        "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/AWSBatchJobRole*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"arn:aws:iam::*:role/*Batch*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "batch.amazonaws.com"
```

```
            }
        }
      }
    ]
  }
```

AWS Batch Aktualisierungen AWS verwalteter Richtlinien

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien AWS Batch seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst. Abonnieren Sie den RSS-Feed auf der Seite AWS Batch Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderung	Beschreibung	Datum
BatchServiceRolePolicyRichtlinie aktualisiert	Es wurde aktualisiert und bietet nun Unterstützung für die Beschreibung des Anforderungsverlaufs und der Amazon EC2 Auto Scaling Aktivitäten von Spot Fleet.	05. Dezember 20
AWSBatchServiceRoleRichtlinie hinzugefügt	Es wurde aktualisiert, um eine Erklärung hinzuzufügen IDs, AWS Batch Berechtigungen zu gewähren `ec2:DescribeSpotFleetRequestHistory` und `autoscaling:DescribeScalingActivities` .	05. Dezember 20
BatchServiceRolePolicyRichtlinie aktualisiert	Aktualisiert, um Unterstützung für die Beschreibung von Amazon EKS-Clustern hinzuzufügen.	20. Oktober 2022
AWSBatchFullAccessRichtlinie aktualisiert	Aktualisiert, um Unterstützung für das Auflisten und Beschreiben von Amazon EKS-Clustern hinzuzufügen.	20. Oktober 2022

Änderung	Beschreibung	Datum
BatchServiceRolePolicyRichtlinie aktualisiert	Aktualisiert, um Unterstützung für Amazon EC2 Capacity Reservation-Gruppen hinzuzufügen, die von verwaltet werden AWS Resource Groups. Weitere Informationen finden Sie unter Arbeiten mit Kapazitätsreservierungsgruppen im EC2 Amazon-Benutzerhandbuch.	18. Mai 2022
BatchServiceRolePolicyund AWSBatchServiceRoledie Richtlinien wurden aktualisiert	Es wurde aktualisiert, um Unterstützung für die Beschreibung des Status AWS Batch verwalteter Instances in Amazon hinzuzufügen, EC2 sodass fehlerhafte Instances ersetzt werden.	6. Dezember 2021
BatchServiceRolePolicyRichtlinie aktualisiert	Aktualisiert, um Unterstützung für Platzierungsgruppen-, Kapazitätsreservierungs-, Elastic GPU- und Elastic Inference Inference-Ressourcen in Amazon EC2 hinzuzufügen.	26. März 2021
BatchServiceRolePolicyRichtlinie hinzugefügt	Mit der BatchServiceRolePolicyverwalteten Richtlinie für die AWSServiceRoleForBatchdienstverknüpfte Rolle können Sie eine dienstverknüpfte Rolle verwenden, die von verwaltet wird. AWS Batch Mit dieser Richtlinie müssen Sie Ihre eigene Rolle für die Verwendung in Ihren Computerumgebungen nicht beibehalten.	10. März 2021
AWSBatchFullAccess- Fügen Sie die Berechtigung hinzu, um eine dienstbezogene Rolle hinzuzufügen	Fügen Sie IAM-Berechtigungen hinzu, damit die AWSServiceRoleForBatchdienstverknüpfte Rolle dem Konto hinzugefügt werden kann.	10. März 2021

Änderung	Beschreibung	Datum
AWS Batch hat begonnen, Änderungen zu verfolgen	AWS Batch hat begonnen, Änderunge n für die AWS verwalteten Richtlinien zu verfolgen.	10. März 2021

AWS Batch IAM-Richtlinien, -Rollen und -Berechtigungen

Standardmäßig sind Benutzer nicht berechtigt, AWS Batch Ressourcen zu erstellen oder zu ändern oder Aufgaben mithilfe der AWS Batch API, der AWS Batch Konsole oder der auszuführen AWS CLI. Damit Benutzer diese Aktionen ausführen können, erstellen Sie IAM-Richtlinien, die Benutzern Berechtigungen für die spezifischen Ressourcen und API-Operationen gewähren. Fügen Sie dann die Richtlinien den Benutzern oder Gruppen hinzu, für die diese Berechtigungen erforderlich sind.

Wenn Sie einem Benutzer oder einer Benutzergruppe eine Richtlinie zuordnen, erlaubt oder verweigert die Richtlinie die Berechtigungen zur Ausführung bestimmter Aufgaben für bestimmte Ressourcen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Berechtigungen und Richtlinien. Weitere Informationen zum Verwalten und Erstellen von benutzerdefinierten IAM-Richtlinien finden Sie unter Verwalten von IAM-Richtlinien.

AWS Batch tätigt in Ihrem Namen Anrufe AWS-Services an andere. Daher AWS Batch müssen Sie sich mit Ihren Anmeldeinformationen authentifizieren. Genauer gesagt, AWS Batch authentifiziert sich, indem eine IAM-Rolle und eine IAM-Richtlinie erstellt werden, die diese Berechtigungen bereitstellen. Anschließend ordnet sie die Rolle Ihren Computerumgebungen zu, wenn Sie sie erstellen. Weitere Informationen finden Sie unter Amazon ECS-Instance-Rolle IAM-Rollen, Verwenden von dienstverknüpften Rollen und Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

Themen

- Struktur der IAM-Richtlinien
- Ressource: Beispielrichtlinien für AWS Batch
- Ressource: AWS Batch verwaltete Richtlinie

Struktur der IAM-Richtlinien

In den folgenden Themen wird die Struktur einer IAM-Richtlinie erläutert.

Themen

- Richtliniensyntax
- API-Aktionen für AWS Batch
- Amazon-Ressourcennamen für AWS Batch
- Vergewissern Sie sich, dass Benutzer über die erforderlichen Berechtigungen verfügen

Richtliniensyntax

Eine IAM-Richtlinie ist ein JSON-Dokument, das eine oder mehrere Anweisungen enthält. Jede Anweisung ist folgendermaßen strukturiert.

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
    "key":"value"
    }
      }
    }
  ]
}
```

Eine Aussage besteht aus vier Hauptelementen:

- Effect: Der effect-Wert kann `Allow` oder `Deny` lauten. Standardmäßig sind Benutzer nicht berechtigt, Ressourcen und API-Aktionen zu verwenden. Daher werden alle Anfragen abgelehnt. Dieser Standardwert kann durch eine explizite Zugriffserlaubnis überschrieben werden. Eine explizite Zugriffsverweigerung überschreibt jedwede Zugriffserlaubnis.
- Aktion: Die Aktion ist die spezifische API-Aktion, für die Sie die Erlaubnis erteilen oder verweigern. Anweisungen zur Spezifizierung der Aktion finden Sie unterAPI-Aktionen für AWS Batch.
- Resource: Die von einer Aktion betroffene Ressource. Bei einigen AWS Batch API-Aktionen können Sie bestimmte Ressourcen in Ihre Richtlinie aufnehmen, die durch die Aktion erstellt oder geändert werden können. Um eine Ressource in der Anweisung anzugeben, verwenden Sie deren Amazon-Ressourcennamen (ARN). Weitere Informationen erhalten Sie unter Unterstützte Berechtigungen auf Ressourcenebene für API-Aktionen AWS Batch und

Amazon-Ressourcennamen für AWS Batch. Wenn der AWS Batch API-Vorgang derzeit keine Berechtigungen auf Ressourcenebene unterstützt, fügen Sie einen Platzhalter (*) hinzu, um anzugeben, dass alle Ressourcen von der Aktion betroffen sein können.

- Condition: Bedingungen sind optional. Mit ihrer Hilfe können Sie bestimmen, wann Ihre Richtlinie wirksam ist.

Weitere Informationen zu beispielhaften IAM-Richtlinienanweisungen für finden Sie unter. AWS BatchRessource: Beispielrichtlinien für AWS Batch

API-Aktionen für AWS Batch

In einer IAM-Richtlinienanweisung können Sie jede API-Aktion von jedem Service, der IAM unterstützt, angeben. Verwenden Sie für AWS Batch das folgende Präfix mit dem Namen der API-Aktion: `batch:` (zum Beispiel `batch:SubmitJob` und`batch:CreateComputeEnvironment`).

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie jede Aktion durch ein Komma.

```
"Action": ["batch:action1", "batch:action2"]
```

Sie können auch mehrere Aktionen angeben, indem Sie einen Platzhalter (*) angeben. Sie können beispielsweise alle Aktionen mit einem Namen angeben, der mit dem Wort „Describe“ beginnt.

```
"Action": "batch:Describe*"
```

Um alle AWS Batch API-Aktionen anzugeben, fügen Sie einen Platzhalter (*) hinzu.

```
"Action": "batch:*"
```

Eine Liste der AWS Batch Aktionen finden Sie unter Aktionen in der AWS Batch API-Referenz.

Amazon-Ressourcennamen für AWS Batch

Jede IAM-Richtlinienerklärung gilt für die Ressourcen, die Sie mit ihren Amazon-Ressourcennamen (ARNs) angeben.

Ein Amazon-Ressourcenname (ARN) hat die folgende allgemeine Syntax:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

Service nicht zulässig

Der Service (z. B. `batch`)

Region

Der AWS-Region für die Ressource (zum Beispielus`-east-2`).

Konto

Die AWS-Konto ID ohne Bindestriche (z. B.`123456789012`).

RessourcenTyp

Der Typ der Ressource (z. B. `compute-environment`)

resourcePath

Ein Pfad zur Identifizierung der Ressource. Sie können in Ihren Pfaden einen Platzhalter (*) verwenden.

AWS Batch API-Operationen unterstützen derzeit Berechtigungen auf Ressourcenebene für mehrere API-Operationen. Weitere Informationen finden Sie unter Unterstützte Berechtigungen auf Ressourcenebene für API-Aktionen AWS Batch. Um alle Ressourcen anzugeben oder falls eine bestimmte API-Aktion dies nicht unterstützt ARNs, fügen Sie dem Element einen Platzhalter (*) hinzu. `Resource`

```
"Resource": "*"
```

Vergewissern Sie sich, dass Benutzer über die erforderlichen Berechtigungen verfügen

Bevor Sie eine IAM-Richtlinie in Betrieb nehmen, stellen Sie sicher, dass sie Benutzern die Berechtigungen zur Nutzung der spezifischen API-Aktionen und Ressourcen gewährt, die sie benötigen.

Erstellen Sie dazu zunächst einen Benutzer zu Testzwecken und hängen Sie die IAM-Richtlinie an den Testbenutzer an. Anschließend initiieren Sie mit dem Testbenutzer eine Anforderung. Anforderungen erfolgen über die Konsole oder die AWS CLI.

> **Note**
>
> Sie können Ihre Richtlinien auch mit dem IAM Policy Simulator testen. Weitere Informationen zum Richtliniensimulator finden Sie unter Arbeiten mit dem IAM Policy Simulator im IAM-Benutzerhandbuch.

Falls die Richtlinie dem Benutzer nicht die erwarteten Berechtigungen erteilt oder zu viele Berechtigungen gewährt, können Sie die Richtlinie entsprechend anpassen. Testen Sie so lange, bis Sie die gewünschten Ergebnisse erhalten.

> **Important**
>
> Es kann einige Minuten dauern, bis Richtlinienänderungen wirksam werden. Daher empfehlen wir, dass Sie mindestens fünf Minuten verstreichen lassen, bevor Sie Ihre Richtlinienaktualisierungen testen.

Bei einer fehlgeschlagenen Autorisierungsprüfung gibt die Anforderung eine codierte Nachricht mit Diagnoseinformationen zurück. Sie können die Nachricht mit der Aktion `DecodeAuthorizationMessage` decodieren. Weitere Informationen finden Sie DecodeAuthorizationMessagein der AWS Security Token Service API-Referenz und decode-authorization-messagein der AWS CLI Befehlsreferenz.

Ressource: Beispielrichtlinien für AWS Batch

Sie können spezielle IAM-Richtlinien erstellen, um die Anrufe und Ressourcen einzuschränken, auf die Benutzer in Ihrem Konto Zugriff haben. Anschließend können Sie diese Richtlinien an Benutzer anhängen.

Wenn Sie einem Benutzer oder einer Benutzergruppe eine Richtlinie zuordnen, gewährt oder verweigert die Richtlinie dem Benutzer die Berechtigung für bestimmte Aufgaben auf bestimmten Ressourcen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Berechtigungen und Richtlinien. Anweisungen zur Verwaltung und Erstellung benutzerdefinierter IAM-Richtlinien finden Sie unter Verwaltung von IAM-Richtlinien.

Die folgenden Beispiele zeigen Richtlinienanweisungen, mit denen Sie steuern können, welche Berechtigungen Benutzer haben. AWS Batch

Beispiele

- Ressource: Schreibgeschützter Zugriff für AWS Batch
- Ressource: Beschränken Sie sich bei der Auftragseingabe auf POSIX-Benutzer, Docker-Image, Rechtestufe und Rolle
- Ressource: Beschränken Sie sich bei der Auftragseingabe auf das Jobdefinitionspräfix
- Ressource: Auf eine Job-Warteschlange beschränken
- Aktion ablehnen, wenn alle Bedingungen den Zeichenfolgen entsprechen
- Ressource: Aktion verweigern, wenn Bedingungsschlüssel mit Zeichenketten übereinstimmen
- Ressource: Verwenden Sie den batch:ShareIdentifier Bedingungsschlüssel

Ressource: Schreibgeschützter Zugriff für AWS Batch

Die folgende Richtlinie gewährt Benutzern die Erlaubnis, alle AWS Batch API-Aktionen zu verwenden, deren Name mit und beginnt. `Describe` `List`

Sofern ihnen nicht eine andere Erklärung die Erlaubnis dazu erteilt, sind Benutzer nicht berechtigt, Aktionen mit den Ressourcen durchzuführen. Standardmäßig wird ihnen die Erlaubnis verweigert, API-Aktionen zu verwenden.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:Describe*",
                "batch:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Ressource: Beschränken Sie sich bei der Auftragseingabe auf POSIX-Benutzer, Docker-Image, Rechtestufe und Rolle

Die folgende Richtlinie ermöglicht es einem POSIX-Benutzer, seinen eigenen Satz von eingeschränkten Jobdefinitionen zu verwalten.

Verwenden Sie die erste und die zweite Anweisung, um jeden Jobdefinitionsnamen zu registrieren oder zu deregistrieren, dessen Namen ein Präfix vorangestellt ist. *JobDefA_*

Die erste Anweisung verwendet auch bedingte Kontextschlüssel zum Beschränken des POSIX-Benutzers, des privilegierten Status und der Container-Image-Werte innerhalb der `containerProperties` einer Auftragsdefinition. Weitere Informationen finden Sie unter RegisterJobDefinition in der AWS Batch -API-Referenz. In diesem Beispiel können Jobdefinitionen nur registriert werden, wenn der POSIX-Benutzer auf gesetzt ist. `nobody` Das privilegierte Flag ist auf `false` gesetzt. Zuletzt wird das Bild `myImage` in einem Amazon ECR-Repository gespeichert.

> **Important**
>
> Docker löst den `user` Parameter innerhalb des Container-Images für diesen Benutzer `uid` auf. In den meisten Fällen befindet sich dies in der `/etc/passwd` Datei im Container-Image. Diese Namensauflösung kann vermieden werden, indem direkte `uid` Werte sowohl in der Auftragsdefinition als auch in allen zugehörigen IAM-Richtlinien verwendet werden. Sowohl die AWS Batch API-Operationen als auch die bedingten `batch:User` IAM-Schlüssel unterstützen numerische Werte.

Verwenden Sie die dritte Anweisung, um eine Jobdefinition nur auf eine bestimmte Rolle zu beschränken.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:RegisterJobDefinition"
            ],
```

```
            "Resource": [
                "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/
JobDefA_*"
            ],
            "Condition": {
                "StringEquals": {
                    "batch:User": [
                        "nobody"
                    ],
                    "batch:Image": [
                        "<aws_account_id>.dkr.ecr.<aws_region>.amazonaws.com/
myImage"
                    ]
                },
                "Bool": {
                    "batch:Privileged": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "batch:DeregisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/
JobDefA_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::<aws_account_id>:role/MyBatchJobRole"
            ]
        }
    ]
}
```

Ressource: Beschränken Sie sich bei der Auftragseingabe auf das Jobdefinitionspräfix

Verwenden Sie die folgende Richtlinie, um Jobs an jede Auftragswarteschlange mit einem beliebigen Jobdefinitionsnamen zu senden, der mit beginnt*JobDefA*.

> **Important**
>
> Beim Festlegen des Zugriffs für das Absenden von Aufträgen auf Ressourcenebene müssen Sie die Ressourcentypen der Auftragswarteschlange und der Auftragsdefinition angeben.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/*"
      ]
    }
  ]
}
```

Ressource: Auf eine Job-Warteschlange beschränken

Verwenden Sie die folgende Richtlinie, um Jobs an eine bestimmte Auftragswarteschlange mit dem Namen Queue1 mit einem beliebigen Auftragsdefinitionsnamen weiterzuleiten.

> **Important**
>
> Beim Festlegen des Zugriffs für das Absenden von Aufträgen auf Ressourcenebene müssen Sie die Ressourcentypen der Auftragswarteschlange und der Auftragsdefinition angeben.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/queue1"
      ]
    }
  ]
}
```

Aktion ablehnen, wenn alle Bedingungen den Zeichenfolgen entsprechen

Die folgende Richtlinie verweigert den Zugriff auf den RegisterJobDefinitionAPI-Vorgang, wenn sowohl der Bedingungsschlüssel `batch:Image` (Container-Image-ID) "*string1*" als auch der Bedingungsschlüssel `batch:LogDriver` (Container-Protokolltreiber) "" lautet*string2*. AWS Batch wertet Bedingungsschlüssel für jeden Container aus. Wenn sich ein Job über mehrere Container erstreckt, z. B. ein parallel Job mit mehreren Knoten, ist es möglich, dass die Container unterschiedliche Konfigurationen haben. Wenn mehrere Bedingungsschlüssel in einer Anweisung ausgewertet werden, werden sie mithilfe von `AND` Logik kombiniert. Wenn also einer der mehreren Bedingungsschlüssel nicht zu einem Container passt, wird der `Deny` Effekt nicht auf diesen Container angewendet. Vielmehr könnte ein anderer Container im selben Job verweigert werden.

Eine Liste der Bedingungsschlüssel für AWS Batch finden Sie unter Bedingungsschlüssel für AWS Batch in der Service Authorization Reference. Mit Ausnahme `batch:ShareIdentifier` von können alle `batch` Bedingungsschlüssel auf diese Weise verwendet werden. Der `batch:ShareIdentifier` Bedingungsschlüssel ist für einen Job definiert, nicht für eine Jobdefinition.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": "batch:RegisterJobDefinition",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "batch:Image": "string1",
          "batch:LogDriver": "string2"
        }
      }
    }
  ]
}
```

Ressource: Aktion verweigern, wenn Bedingungsschlüssel mit Zeichenketten übereinstimmen

Die folgende Richtlinie verweigert den Zugriff auf den RegisterJobDefinitionAPI-Vorgang, wenn entweder der Bedingungsschlüssel `batch:Image` (Container-Image-ID) "*string1*" oder der Bedingungsschlüssel `batch:LogDriver` (Container-Protokolltreiber) "" lautet*string2*. Wenn sich ein Job über mehrere Container erstreckt, z. B. ein parallel Job mit mehreren Knoten, ist es möglich, dass die Container unterschiedliche Konfigurationen haben. Wenn mehrere Bedingungsschlüssel in einer Anweisung ausgewertet werden, werden sie mithilfe von `AND` Logik kombiniert. Wenn also einer der mehreren Bedingungsschlüssel nicht zu einem Container passt, wird der `Deny` Effekt nicht

auf diesen Container angewendet. Vielmehr könnte ein anderer Container im selben Job verweigert werden.

Eine Liste der Bedingungsschlüssel für AWS Batch finden Sie unter Bedingungsschlüssel für AWS Batch in der Service Authorization Reference. Mit Ausnahme `batch:ShareIdentifier` von können alle `batch` Bedingungsschlüssel auf diese Weise verwendet werden. (Der `batch:ShareIdentifier` Bedingungsschlüssel ist für einen Job definiert, nicht für eine Jobdefinition.)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:Image": [
            "string1"
          ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
```

```
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:LogDriver": [
            "string2"
          ]
        }
      }
    }
  ]
}
```

Ressource: Verwenden Sie den **`batch:ShareIdentifier`** Bedingungsschlüssel

Verwenden Sie die folgende Richtlinie, um Jobs, die die `jobDefA` Auftragsdefinition verwenden, mit der `lowCpu` gemeinsamen ID an die `jobqueue1` Auftragswarteschlange zu senden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-definition/JobDefA",
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {
          "batch:ShareIdentifier": [
            "lowCpu"
          ]
        }
```

```
      }
    }
  ]
}
```

Ressource: AWS Batch verwaltete Richtlinie

AWS Batch stellt eine verwaltete Richtlinie bereit, die Sie Benutzern zuordnen können. Diese Richtlinie gewährt die Erlaubnis zur Nutzung von AWS Batch Ressourcen und API-Vorgängen. Sie können diese Richtlinie direkt anwenden oder als Ausgangspunkt zur Erstellung eigener Richtlinien verwenden. Weitere Informationen zu den einzelnen API-Vorgängen, die in diesen Richtlinien erwähnt werden, finden Sie in der AWS Batch API-Referenz unter Aktionen.

AWSBatchFullAccess

Diese Richtlinie ermöglicht vollen Administratorzugriff auf AWS Batch.

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
```

```
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource":"*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":[
        "arn:aws:iam::*:role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/ecsInstanceRole",
        "arn:aws:iam::*:instance-profile/ecsInstanceRole",
        "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/AWSBatchJobRole*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"arn:aws:iam::*:role/*Batch*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "batch.amazonaws.com"
          }
      }
    }
  ]
}
```

AWS Batch IAM-Ausführungsrolle

Die Ausführungsrolle erteilt dem Amazon ECS-Container und den AWS Fargate Agenten die Erlaubnis, AWS API-Aufrufe in Ihrem Namen durchzuführen.

> **Note**
>
> Die Ausführungsrolle wird vom Amazon ECS-Container-Agenten Version 1.16.0 und höher unterstützt.

Die IAM-Ausführungsrolle ist je nach den Anforderungen Ihrer Aufgabe erforderlich. Sie können Ihrem Konto mehrere Ausführungsrollen für unterschiedliche Zwecke und Dienste zuordnen.

> **Note**
>
> Informationen zur Amazon ECS-Instance-Rolle finden Sie unterAmazon ECS-Instance-Rolle. Informationen zu Servicerollen finden Sie unterWie AWS Batch funktioniert mit IAM.

Amazon ECS stellt die `AmazonECSTaskExecutionRolePolicy` verwaltete Richtlinie bereit. Diese Richtlinie enthält die erforderlichen Berechtigungen für die oben beschriebenen allgemeinen Anwendungsfälle. Für die unten beschriebenen speziellen Anwendungsfälle kann es erforderlich sein, Ihrer Ausführungsrolle Inline-Richtlinien hinzuzufügen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

Unterstützte Berechtigungen auf Ressourcenebene für API-Aktionen AWS Batch

Der Begriff Berechtigungen auf Ressourcenebene bezieht sich auf die Möglichkeit, die Ressourcen anzugeben, für die Benutzer Aktionen ausführen dürfen. AWS Batch unterstützt teilweise Berechtigungen auf Ressourcenebene. Bei einigen AWS Batch Aktionen können Sie auf der Grundlage von Bedingungen, die erfüllt sein müssen, steuern, wann Benutzer diese Aktionen verwenden dürfen. Sie können die Steuerung auch auf der Grundlage der spezifischen Ressourcen vornehmen, die Benutzer verwenden dürfen. Zum Beispiel können Sie Benutzern die Berechtigungen dazu erteilen, Aufträge zu übergeben, jedoch nur an eine bestimmte Auftragswarteschlange und nur mit einer bestimmten Auftragsdefinition.

Einzelheiten zu den von AWS Batch definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Batch in der Service Authorization Reference.

Tutorial: Erstellen Sie die IAM-Ausführungsrolle

Wenn Ihr Konto noch nicht über eine IAM-Ausführungsrolle verfügt, verwenden Sie die folgenden Schritte, um die Rolle zu erstellen.

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS-Service.
5. Wählen Sie für Service oder Anwendungsfall Elastic Container Service aus. Wählen Sie dann erneut Elastic Container Service Task.
6. Wählen Sie Weiter aus.
7. Suchen Sie nach Richtlinien für Berechtigungen nach Amazon ECSTask ExecutionRolePolicy.
8. Aktivieren Sie das Kontrollkästchen links neben der ECSTaskExecutionRolePolicyAmazon-Richtlinie und wählen Sie dann Weiter aus.
9. Geben Sie als Rollenname den Text Rolle erstellen ein **`ecsTaskExecutionRole`** und wählen Sie dann Create role aus.

Tutorial: Überprüfen Sie die IAM-Ausführungsrolle

Gehen Sie wie folgt vor, um zu überprüfen, ob Ihr Konto bereits über die IAM-Ausführungsrolle verfügt, und fügen Sie gegebenenfalls die verwaltete IAM-Richtlinie hinzu.

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Suchen Sie in der Liste der Rollen nach `ecsTaskExecutionRole`. Wenn Sie die Rolle nicht finden können, finden Sie weitere Informationen unter. Tutorial: Erstellen Sie die IAM-Ausführungsrolle Wenn Sie die Rolle gefunden haben, wählen Sie die Rolle aus, um die angehängten Richtlinien einzusehen.
4. Stellen Sie auf der Registerkarte Berechtigungen sicher, dass die von Amazon ECSTask ExecutionRolePolicy verwaltete Richtlinie mit der Rolle verknüpft ist. Wenn die Richtlinie angehängt ist, ist Ihre Ausführungsrolle ordnungsgemäß konfiguriert. Andernfalls führen Sie die folgenden Teilschritte aus, um die Richtlinie zuzuweisen.
   a. Wählen Sie „Berechtigungen hinzufügen“ und anschließend „Richtlinien anhängen“.
   b. Suchen Sie nach Amazon ECSTask ExecutionRolePolicy.
   c. Markieren Sie das Kästchen links neben der ECSTaskExecutionRolePolicyAmazon-Richtlinie und wählen Sie Richtlinien anhängen aus.
5. Wählen Sie Trust Relationships (Vertrauensbeziehungen) aus.
6. Überprüfen Sie, dass die Vertrauensstellung die folgende Richtlinie enthält. Wenn das Vertrauensverhältnis mit der folgenden Richtlinie übereinstimmt, ist die Rolle korrekt konfiguriert. Wenn die Vertrauensstellung nicht übereinstimmt, wählen Sie Vertrauensrichtlinie bearbeiten, geben Sie Folgendes ein und wählen Sie Richtlinie aktualisieren aus.

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
```

```
        "Action": "sts:AssumeRole"
      }
    ]
}
```

Verwenden Sie dienstbezogene Rollen für AWS Batch

AWS Batch verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, mit der direkt verknüpft ist. AWS Batch Mit Diensten verknüpfte Rollen sind vordefiniert AWS Batch und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle AWS Batch erleichtert die Einrichtung, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS Batch definiert die Berechtigungen ihrer dienstbezogenen Rollen und AWS Batch kann, sofern nicht anders definiert, nur ihre Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

> **Note**
>
> Führen Sie einen der folgenden Schritte aus, um eine Servicerolle für eine AWS Batch Rechenumgebung anzugeben.
>
> - Verwenden Sie eine leere Zeichenfolge für die Servicerolle. Dadurch können AWS Batch Sie die Servicerolle erstellen.
> - Geben Sie die Servicerolle im folgenden Format an:`arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.
>
> Weitere Informationen finden Sie Falscher Rollenname oder ARN im AWS Batch Benutzerhandbuch.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre AWS Batch -Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Dienstbezogene Rollenberechtigungen für AWS Batch

AWS Batch verwendet die benannte dienstverknüpfte Rolle. AWSServiceRoleForBatch Die AWSServiceRoleForBatchRolle ermöglicht es AWS Batch , AWS Ressourcen in Ihrem Namen zu erstellen und zu verwalten.

Die serviceverknüpfte AWSServiceRoleForBatch-Rolle vertraut dem `batch.amazonaws.com`-Service-Prinzipal im Rahmen der Übernahme der Rolle.

Die genannte IAM-Richtlinie BatchServiceRolePolicyermöglicht es AWS Batch , die folgenden Aktionen für bestimmte Ressourcen durchzuführen:

- `autoscaling`— Ermöglicht das AWS Batch Erstellen und Verwalten von Amazon EC2 Auto Scaling Scaling-Ressourcen. AWS Batch erstellt und verwaltet Amazon EC2 Auto Scaling Scaling-Gruppen für die meisten Computerumgebungen.
- `ec2`— Ermöglicht AWS Batch die Kontrolle des Lebenszyklus von EC2 Amazon-Instances sowie die Erstellung und Verwaltung von Startvorlagen und Tags. AWS Batch erstellt und verwaltet EC2 Spot-Flottenanfragen für einige EC2 Spot-Computerumgebungen.
- `ecs`- Ermöglicht AWS Batch die Erstellung und Verwaltung von Amazon ECS-Clustern, Aufgabendefinitionen und Aufgaben für die Auftragsausführung.
- `eks`- Ermöglicht AWS Batch die Beschreibung der Amazon EKS-Cluster-Ressource für Validierungen.
- `iam`- Ermöglicht AWS Batch die Validierung und Weitergabe von Rollen, die vom Eigentümer bereitgestellt wurden, an Amazon EC2, Amazon EC2 Auto Scaling und Amazon ECS.
- `logs`— Ermöglicht AWS Batch das Erstellen und Verwalten von Protokollgruppen und Protokollströmen für AWS Batch Jobs.

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter serviceverknüpfte Rollenberechtigungen im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für AWS Batch

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie sich CreateComputeEnvironment in der AWS Management Console AWS CLI, der oder der AWS API befinden und keinen Wert für den `serviceRole` Parameter angeben, AWS Batch wird die dienstbezogene Rolle für Sie erstellt.

> **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. Außerdem, wenn Sie den AWS Batch Dienst vor dem 10. März 2021 genutzt haben, als er begann, dienstbezogene Rollen zu unterstützen, dann AWS Batch haben Sie die AWSService RoleForBatch Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter Eine neue Rolle ist in meinem IAM-Konto erschienen.

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 CreateComputeEnvironment, AWS Batch erstellt die dienstbezogene Rolle erneut für Sie.

Bearbeitung einer serviceverknüpften Rolle für AWS Batch

Mit AWS Batch können Sie die AWSService RoleForBatch dienstverknüpfte Rolle nicht bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Um einer IAM-Entität zu ermöglichen, die Beschreibung der serviceverknüpften Rolle zu bearbeiten AWSService RoleForBatch

Fügen Sie der Berechtigungsrichtlinie die folgende Anweisung hinzu. Dadurch kann die IAM-Entität die Beschreibung einer serviceverknüpften Rolle bearbeiten.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
```

```
    "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
    "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Löschen einer serviceverknüpften Rolle für AWS Batch

Wir empfehlen, dass Sie diese Rolle löschen, wenn Sie eine Funktion oder einen Dienst nicht mehr verwenden müssen, für den eine dienstverknüpfte Rolle erforderlich ist. Auf diese Weise verfügen Sie nicht über eine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Um einer IAM-Entität zu ermöglichen, die mit dem AWSService RoleForBatch Service verknüpfte Rolle zu löschen

Fügen Sie der Berechtigungsrichtlinie die folgende Anweisung hinzu. Dadurch kann die IAM-Entität eine dienstverknüpfte Rolle löschen.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
    "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Bereinigen einer serviceverknüpften Rolle

Bevor Sie IAM verwenden können, um eine dienstverknüpfte Rolle zu löschen, müssen Sie zunächst bestätigen, dass die Rolle keine aktiven Sitzungen hat, und alle AWS Batch Computerumgebungen, die die Rolle verwenden, in allen AWS Regionen in einer einzigen Partition löschen.

So überprüfen Sie, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen und dann den AWSService RoleForBatch Namen aus (nicht das Kontrollkästchen).

3. Wählen Sie auf der Seite Summary Access Advisor und überprüfen Sie die letzten Aktivitäten auf die serviceverknüpfte Rolle.

> **Note**
>
> Wenn Sie nicht wissen, ob die AWSService RoleForBatch Rolle verwendet AWS Batch wird, können Sie versuchen, die Rolle zu löschen. Wenn der Dienst die Rolle verwendet, kann die Rolle nicht gelöscht werden. Sie können die Regionen anzeigen, in denen die Rolle verwendet wird. Wenn die Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet wird, bevor Sie die Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Um AWS Batch Ressourcen zu entfernen, die von der AWSService RoleForBatch serviceverknüpften Rolle verwendet werden

Sie müssen alle AWS Batch Rechenumgebungen, die die AWSService RoleForBatch Rolle verwenden, in allen AWS Regionen löschen, bevor Sie die AWSService RoleForBatch Rolle löschen können.

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie die zu verwendende Region in der Navigationsleiste aus.
3. Wählen Sie im Navigationsbereich Datenverarbeitungs-Umgebungen aus.
4. Wählen Sie die Rechenumgebung aus.
5. Wählen Sie Disable (deaktivieren) aus. Warten Sie, bis der Status auf DEAKTIVIERT wechselt.
6. Wählen Sie die Rechenumgebung aus.
7. Wählen Sie Löschen aus. Bestätigen Sie, dass Sie die Rechenumgebung löschen möchten, indem Sie Rechenumgebung löschen wählen.
8. Wiederholen Sie die Schritte 1—7 für alle Rechenumgebungen, die die serviceverknüpfte Rolle verwenden, in allen Regionen.

Löschen einer serviceverknüpften Rolle in IAM (Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

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 im Navigationsbereich der IAM Console Roles (Rollen) aus. Aktivieren Sie dann das Kontrollkästchen neben AWSServiceRoleForBatch, nicht den Namen oder die Zeile selbst.
3. Wählen Sie Delete role (Rolle löschen) aus.
4. Überprüfen Sie im Bestätigungsdialogfeld die letzten Service-Zugriffsdaten, die zeigen, wann jede der ausgewählten Rollen zuletzt auf einen AWS-Service zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wenn Sie fortfahren möchten, wählen Sie Yes, Delete aus, um die serviceverknüpfte Rolle zur Löschung zu übermitteln.
5. Sehen Sie sich die Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da die Löschung der serviceverknüpften IAM-Rolle asynchron erfolgt, kann die Löschung nach dem Übermitteln der Rolle für die Löschung erfolgreich sein oder fehlschlagen.

   - Wenn der Vorgang erfolgreich ist, wird die Rolle aus der Liste entfernt und eine Benachrichtigung des Erfolgs oben auf der Seite angezeigt.
   - Wenn der Vorgang fehlschlägt, können Sie in den Benachrichtigungen View details oder View Resources auswählen, um zu erfahren, warum die Löschung fehlgeschlagen ist. Wenn die Löschung fehlschlägt, weil die Rolle Ressourcen des Services verwendet, enthält die Benachrichtigung eine Liste der Ressourcen – sofern der Service diese Informationen zurückgibt. Sie können dann die Ressourcen bereinigen und die Löschung erneut durchführen.

     > **Note**
     >
     > Möglicherweise müssen Sie diesen Vorgang abhängig von den Informationen, die der Service zurückgibt, mehrmals wiederholen. Ihre serviceverknüpfte Rolle könnte beispielsweise sechs Ressourcen verwenden und Ihr Service Informationen zu fünf davon zurückgeben. Wenn Sie die fünf Ressourcen bereinigen und die Rolle erneut für den Löschvorgang übermitteln, schlägt die Löschung fehl und der Service gibt die eine verbleibende Ressource zurück. Ein Service kann alle Ressourcen zurückgeben oder nur einige bzw. gar keine.

   - Wenn die Aufgabe fehlschlägt und die Benachrichtigung keine Liste der Ressourcen enthält, gibt der Service möglicherweise diese Informationen nicht zurück. Informationen zum Bereinigen von Ressourcen für diesen Service finden Sie unter AWS-Services , die mit

IAM funktionieren. Um die Dokumentation der serviceverknüpften Rolle für den Service anzuzeigen, suchen Sie Ihren Service in der Tabelle und wählen Sie den Link Yes.

Löschen einer serviceverknüpften Rolle in IAM ()AWS CLI

Sie können IAM-Befehle von verwenden, um eine dienstverknüpfte AWS Command Line Interface Rolle zu löschen.

So löschen Sie eine serviceverknüpfte Rolle (CLI)

1. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen. Geben Sie den folgenden Befehl ein, um eine Löschanforderung für eine serviceverknüpfte Rolle zu übermitteln:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
```

2. Verwenden Sie den folgenden Befehl, um den Status der Löschaufgabe zu überprüfen:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können. Wenn die Löschung fehlschlägt, weil die Rolle Ressourcen des Services verwendet, enthält die Benachrichtigung eine Liste der Ressourcen – sofern der Service diese Informationen zurückgibt. Sie können dann die Ressourcen bereinigen und die Löschung erneut durchführen.

> **Note**
>
> Möglicherweise müssen Sie diesen Vorgang abhängig von den Informationen, die der Service zurückgibt, mehrmals wiederholen. Ihre serviceverknüpfte Rolle könnte beispielsweise sechs Ressourcen verwenden und Ihr Service Informationen zu fünf davon zurückgeben. Wenn Sie die fünf Ressourcen bereinigen und die Rolle erneut für den Löschvorgang übermitteln, schlägt die Löschung fehl und der Service gibt die eine

verbleibende Ressource zurück. Ein Dienst kann alle Ressourcen zurückgeben, einige davon. Oder es meldet möglicherweise keine Ressourcen. Informationen zum Bereinigen der Ressourcen für einen Dienst, der keine Ressourcen meldet, finden Sie unter AWS Dienste, die mit IAM funktionieren. Um die Dokumentation der serviceverknüpften Rolle für den Service anzuzeigen, suchen Sie Ihren Service in der Tabelle und wählen Sie den Link Yes.

Löschen einer dienstverknüpften Rolle in IAM (API)AWS

Sie können die IAM-API für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Um eine Löschanfrage für eine serviceverknüpfte Rolle zu übermitteln, rufen Sie DeleteServiceLinkedRole auf. Geben Sie in der Anfrage den AWSService RoleForBatch Rollennamen an.

   Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschungsanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `DeletionTaskId` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.
2. Um den Status der Löschung zu überprüfen, rufen Sie GetServiceLinkedRoleDeletionStatus auf. Geben Sie in der Anforderung die `DeletionTaskId` an.

   Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können. Wenn die Löschung fehlschlägt, weil die Rolle Ressourcen des Services verwendet, enthält die Benachrichtigung eine Liste der Ressourcen – sofern der Service diese Informationen zurückgibt. Sie können dann die Ressourcen bereinigen und die Löschung erneut durchführen.

   > **Note**
   >
   > Möglicherweise müssen Sie diesen Vorgang abhängig von den Informationen, die der Service zurückgibt, mehrmals wiederholen. Ihre serviceverknüpfte Rolle könnte beispielsweise sechs Ressourcen verwenden und Ihr Service Informationen zu fünf davon zurückgeben. Wenn Sie die fünf Ressourcen bereinigen und die Rolle erneut

für den Löschvorgang übermitteln, schlägt die Löschung fehl und der Service gibt die eine verbleibende Ressource zurück. Ein Service kann alle Ressourcen zurückgeben oder nur einige bzw. gar keine. Informationen zum Bereinigen von Ressourcen für einen Service, der keine Ressourcen meldet, finden Sie unter AWS-Services , die mit IAM funktionieren. Um die Dokumentation der serviceverknüpften Rolle für den Service anzuzeigen, suchen Sie Ihren Service in der Tabelle und wählen Sie den Link Yes.

Unterstützte Regionen für AWS Batch serviceverknüpfte Rollen

AWS Batch unterstützt die Verwendung von dienstbezogenen Rollen in allen Regionen, in denen der Dienst verfügbar ist. Weitere Informationen finden Sie unter AWS Batch -Endpunkte.

Amazon ECS-Instance-Rolle

AWS Batch Rechenumgebungen sind mit Amazon ECS-Container-Instances gefüllt. Sie führen den Amazon ECS-Container-Agenten lokal aus. Der Amazon ECS-Container-Agent ruft in Ihrem Namen verschiedene AWS API-Operationen auf. Daher benötigen Container-Instances, auf denen der Agent ausgeführt wird, eine IAM-Richtlinie und eine IAM-Rolle, damit diese Services erkennen, dass der Agent Ihnen gehört. Sie müssen eine IAM-Rolle und ein Instance-Profil erstellen, damit die Container-Instances sie beim Start verwenden können. Andernfalls können Sie keine Rechenumgebung erstellen und Container-Instances darin starten. Diese Anforderung gilt für Container-Instances, die mit oder ohne das von Amazon bereitgestellte Amazon ECS-optimierte AMI gestartet wurden. Weitere Informationen finden Sie unter Amazon ECS-Instance-Rolle im Entwicklerhandbuch für Amazon Elastic Container Service.

Themen

- Tutorial: Überprüfen Sie die Amazon ECS-Instance-Rolle Ihres Kontos

Tutorial: Überprüfen Sie die Amazon ECS-Instance-Rolle Ihres Kontos

Die Amazon ECS-Instance-Rolle und das Instance-Profil werden in der ersten Ausführung der Konsole automatisch für Sie erstellt. Sie können jedoch diesen Schritten folgen, um zu überprüfen, ob Ihr Konto bereits über die Amazon ECS-Instance-Rolle und das Instance-Profil verfügt. In den folgenden Schritten wird auch beschrieben, wie Sie die verwaltete IAM-Richtlinie anhängen.

Tutorial: Suchen Sie **`ecsInstanceRole`** in der IAM-Konsole nach

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Suchen Sie in der Liste der Rollen nach `ecsInstanceRole`. Wenn die Rolle nicht existiert, gehen Sie wie folgt vor, um die Rolle zu erstellen.
   a. Wählen Sie Rolle erstellen aus.
   b. Wählen Sie für Vertrauenswürdige Entität die Option AWS-Service aus.
   c. Wählen Sie für Allgemeine Anwendungsfälle die Option EC2.
   d. Wählen Sie Weiter aus.
   e. Suchen Sie für Berechtigungsrichtlinien nach Amazon EC2 ContainerServicefor EC2 Role.
   f. Aktivieren Sie das Kontrollkästchen neben Amazon EC2 ContainerServicefor EC2 Role und wählen Sie dann Weiter.
   g. Geben Sie unter Role Name (Rollenname) den Namen `ecsInstanceRole` ein und wählen Sie Create role (Rolle erstellen) aus.

   > **Note**
   > Wenn Sie die verwenden, AWS Management Console um eine Rolle für Amazon zu erstellen EC2, erstellt die Konsole ein Instance-Profil mit demselben Namen wie die Rolle.

Alternativ können Sie die verwenden, AWS CLI um die `ecsInstanceRole` IAM-Rolle zu erstellen. Im folgenden Beispiel wird eine IAM-Rolle mit einer Vertrauensrichtlinie und einer AWS verwalteten Richtlinie erstellt.

Tutorial: Erstellen Sie eine IAM-Rolle und ein Instanzprofil ()AWS CLI

1. Erstellen Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Textdatei mit dem Namen`ecsInstanceRole-role-trust-policy.json`.

   JSON

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Verwenden Sie den Befehl create-role, um die `ecsInstanceRole` Rolle zu erstellen. Geben Sie den Speicherort der Vertrauensrichtlinien-Datei im Parameter an`assume-role-policy-document`.

```
$ aws iam create-role \
    --role-name ecsInstanceRole \
    --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "Role": {
         "Path": "/",
         "RoleName: "ecsInstanceRole",
         "RoleId": "AROAT46P5RDIY4EXAMPLE",
         "Arn": "arn:aws:iam::123456789012:role/ecsInstanceRole".
        "CreateDate": "2022-12-12T23:46:37.247Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service: "ec2.amazonaws.com"
                    }
                    "Action": "sts:AssumeRole",
                 }
              ]
          }
        }
```

3. Verwenden Sie den create-instance-profileBefehl, um ein benanntes Instanzprofil zu erstellen`ecsInstanceRole`.

> **Note**
>
> Sie müssen Rollen und Instanzprofile als separate Aktionen in der AWS CLI AWS AND-API erstellen.

```
$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole
```

Nachfolgend finden Sie eine Beispielantwort.

```
{
    "InstanceProfile": {
        "Path": "/",
        "InstanceProfileName": "ecsInstanceRole",
        "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
        "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
        "CreateDate": "2022-06-30T23:53:34.093Z",
        "Roles": [],    }
}
```

4. Verwenden Sie den Befehl add-role-to-instance-profile, um die `ecsInstanceRole` Rolle zum `ecsInstanceRole` Instanzprofil hinzuzufügen.

```
aws iam add-role-to-instance-profile \
    --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Verwenden Sie den attach-role-policyBefehl, um die `AmazonEC2ContainerServiceforEC2Role` AWS verwaltete Richtlinie an die `ecsInstanceRole` Rolle anzuhängen.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
    --role-name ecsInstanceRole
```

Rolle bei Amazon EC2 Spot Fleet

Wenn Sie eine verwaltete Rechenumgebung erstellen, die Amazon EC2 Spot Fleet Instances verwendet, müssen Sie die `AmazonEC2SpotFleetTaggingRole` Richtlinie erstellen. Diese Richtlinie erteilt Spot Fleet die Erlaubnis, Instances in Ihrem Namen zu starten, zu kennzeichnen und zu beenden. Legen Sie die Rolle in Ihrer Spot-Flottenanforderung fest. Sie müssen außerdem über die Rollen AWSServiceRoleForEC2Spot und AWSServiceRoleForEC2SpotFleetserviceverknüpfte Rollen für Amazon EC2 Spot und Spot Fleet verfügen. Verwenden Sie die folgende Anleitung, um all diese Rollen zu erstellen. Weitere Informationen finden Sie unter Verwenden von dienstbezogenen Rollen und Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst im IAM-Benutzerhandbuch.

Themen

- Tutorial: Erstellen Sie Amazon EC2 Spot-Flottenrollen in der AWS Management Console
- Tutorial: Erstellen Sie Amazon EC2 Spot-Flottenrollen mit dem AWS CLI

Tutorial: Erstellen Sie Amazon EC2 Spot-Flottenrollen in der AWS Management Console

So erstellen Sie die serviceverknüpfte **`AmazonEC2SpotFleetTaggingRole`** IAM-Rolle für Amazon EC2 Spot Fleet

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie für Access Management die Option Rollen,
3. Wählen Sie für Rollen die Option Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdige Entität auswählen für Vertrauenswürdigen Entitätstyp die Option aus AWS-Service.
5. Wählen Sie unter „Anwendungsfälle für andere AWS-Services“ die Option EC2und anschließend „EC2 Spot-Fleet-Tagging“ aus.
6. Wählen Sie Weiter aus.
7. Überprüfen **`AmazonEC2SpotFleetTaggingRole`**Sie unter Berechtigungsrichtlinien den Richtliniennamen.
8. Wählen Sie Weiter aus.
9. Überprüfen und erstellen Sie für Name:

a. Geben Sie unter Rollenname einen Namen ein, um die Rolle zu identifizieren.
b. Geben Sie unter Beschreibung eine kurze Erklärung für die Richtlinie ein.
c. (Optional) Wählen Sie für Schritt 1: Vertrauenswürdige Entitäten auswählen und Bearbeiten aus, um den Code zu ändern.
d. (Optional) Wählen Sie für Schritt 2: Berechtigungen hinzufügen die Option Bearbeiten aus, um den Code zu ändern.
e. (Optional) Wählen Sie unter Tags hinzufügen die Option Tag hinzufügen aus, um der Ressource Tags hinzuzufügen.
f. Wählen Sie Rolle erstellen aus.

> **Note**
>
> In der Vergangenheit gab es zwei verwaltete Richtlinien für die Amazon EC2 Spot-Flottenrolle.
>
> - Amazon EC2 SpotFleetRole: Dies ist die ursprünglich verwaltete Richtlinie für die Spot-Flottenrolle. Wir empfehlen jedoch nicht mehr, sie zusammen mit zu verwenden AWS Batch. Diese Richtlinie unterstützt kein Spot-Instance-Tagging in Rechenumgebungen, das für die Verwendung der `AWSServiceRoleForBatch` serviceverknüpften Rolle erforderlich ist. Wenn Sie zuvor eine Spot-Flottenrolle mit dieser Richtlinie erstellt haben, wenden Sie die neue empfohlene Richtlinie auf diese Rolle an. Weitere Informationen finden Sie unter Spot-Instances wurden bei der Erstellung nicht markiert.
> - Amazon EC2 SpotFleetTaggingRole: Diese Rolle bietet alle erforderlichen Berechtigungen, um Amazon EC2 Spot-Instances zu taggen. Verwenden Sie diese Rolle, um Spot-Instance-Tagging in Ihren AWS Batch Computerumgebungen zu ermöglichen.

Tutorial: Erstellen Sie Amazon EC2 Spot-Flottenrollen mit dem AWS CLI

So erstellen Sie die Amazon EC2 SpotFleetTaggingRole IAM-Rolle für Ihre Spot-Flotte-Rechenumgebungen

1. Führen Sie den folgenden Befehl mit dem AWS CLI aus.

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
    --assume-role-policy-document '{
```

```
    "Version":"2012-10-17",
    "Statement":[
       {
         "Sid":"",
         "Effect":"Allow",
         "Principal": {
           "Service":"spotfleet.amazonaws.com"
          },
          "Action":"sts:AssumeRole"
        }
      ]
    }'
```

2. Um die von Amazon EC2 SpotFleetTaggingRole verwaltete IAM-Richtlinie an Ihre EC2SpotFleetTaggingRoleAmazon-Rolle anzuhängen, führen Sie den folgenden Befehl mit dem AWS CLI aus.

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

So erstellen Sie die serviceverknüpfte **`AWSServiceRoleForEC2Spot`** IAM-Rolle für Amazon Spot EC2

> **Note**
>
> Wenn die serviceverknüpfte `AWSServiceRoleForEC2Spot` IAM-Rolle bereits existiert, wird Ihnen eine Fehlermeldung angezeigt, die der folgenden ähnelt.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
>  operation:
> Service role name AWSServiceRoleForEC2Spot has been taken in this account,
>  please try a different suffix.
> ```

- Führen Sie den folgenden Befehl mit dem aus. AWS CLI

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

So erstellen Sie die serviceverknüpfte **`AWSServiceRoleForEC2SpotFleet`** IAM-Rolle für Amazon EC2 Spot Fleet

> **Note**
>
> Wenn die serviceverknüpfte `AWSServiceRoleForEC2SpotFleet` IAM-Rolle bereits existiert, wird Ihnen eine Fehlermeldung angezeigt, die der folgenden ähnelt.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
>  operation:
> Service role name AWSServiceRoleForEC2SpotFleet has been taken in this account,
>  please try a different suffix.
> ```

- Führen Sie den folgenden Befehl mit dem aus. AWS CLI

```
$ aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

EventBridge IAM-Rolle

Amazon EventBridge liefert nahezu in Echtzeit einen Stream von Systemereignissen, die Änderungen an AWS Ressourcen beschreiben. AWS Batch Jobs sind als EventBridge Ziele verfügbar. Mit einfachen, schnell eingerichteten Regeln können Sie Ereignisse zuordnen und in Reaktion darauf AWS Batch -Aufträge ausführen. Bevor Sie AWS Batch Jobs mit EventBridge Regeln und Zielen einreichen können, EventBridge müssen Sie über die erforderlichen Berechtigungen verfügen, um AWS Batch Jobs in Ihrem Namen auszuführen.

> **Note**
>
> Wenn Sie in der EventBridge Konsole eine Regel erstellen, die eine AWS Batch Warteschlange als Ziel angibt, können Sie diese Rolle erstellen. Ein Beispiel-Walkthrough finden Sie unter AWS Batch Jobs als EventBridge Ziele. Sie können die EventBridge Rolle manuell mithilfe der IAM-Konsole erstellen. Anweisungen finden

Sie im IAM-Benutzerhandbuch unter Erstellen einer Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien (Konsole).

Die Vertrauensstellung für Ihre EventBridge IAM-Rolle muss dem `events.amazonaws.com` Dienstprinzipal die Möglichkeit geben, die Rolle zu übernehmen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Stellen Sie sicher, dass die mit Ihrer EventBridge IAM-Rolle verknüpfte Richtlinie `batch:SubmitJob` Berechtigungen für Ihre Ressourcen zulässt. Im folgenden Beispiel wird die `AWSBatchServiceEventTargetRole` verwaltete Richtlinie zur Bereitstellung dieser Berechtigungen bereitgestellt. AWS Batch

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
       ],
      "Resource": "*"
```

```
    }
  ]
}
```

Erstellen einer Virtual Private Cloud

Rechenressourcen in Ihren Datenverarbeitungsumgebungen benötigen externen Netzwerkzugriff, um mit AWS Batch Amazon ECS-Serviceendpunkten kommunizieren zu können. Möglicherweise haben Sie jedoch Jobs, die Sie in privaten Subnetzen ausführen möchten. Um die Flexibilität zu haben, Jobs entweder in einem öffentlichen oder privaten Subnetz auszuführen, erstellen Sie eine VPC, die sowohl öffentliche als auch private Subnetze hat.

Sie können Amazon Virtual Private Cloud (Amazon VPC) verwenden, um AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk zu starten. Dieses Thema enthält einen Link zum Amazon VPC-Assistenten und eine Liste der auszuwählenden Optionen.

Erstellen einer VPC

Informationen zum Erstellen einer Amazon VPC finden Sie unter Nur VPC erstellen im Amazon VPC-Benutzerhandbuch. Ermitteln Sie anhand der folgenden Tabelle, welche Optionen Sie auswählen sollten.

Option	Wert
Zu erstellende Ressourcen	Nur VPC
Name	Geben Sie optional einen Namen für Ihre VPC ein.
IPv4 CIDR-Block	IPv4 Manuelle CIDR-Eingabe Die CIDR-Blockgröße muss eine Größe zwischen /16 und /28. haben.
IPv6 CIDR-Block	Kein IPv6 CIDR-Block
Tenancy	Standard

Weitere Informationen über Amazon VPC finden Sie unter Was ist Amazon VPC? im Amazon VPC-Benutzerhandbuch.

Nächste Schritte

Nachdem Sie Ihre VPC erstellt haben, sollten Sie die folgenden nächsten Schritte in Betracht ziehen:

- Erstellen von Sicherheitsgruppen für Ihre öffentlichen und privaten Ressourcen, wenn diese eingehenden Netzwerkzugriff benötigen. Weitere Informationen finden Sie unter Arbeiten mit Sicherheitsgruppen im Amazon-VPC-Benutzerhandbuch.
- Erstellen Sie eine AWS Batch verwaltete Rechenumgebung, die Rechenressourcen in Ihre neue VPC einführt. Weitere Informationen finden Sie unter Erstellen Sie eine Rechenumgebung. Wenn Sie den Assistenten zur Erstellung einer Rechenumgebung in der AWS Batch Konsole verwenden, können Sie die VPC angeben, die Sie gerade erstellt haben, und die öffentlichen oder privaten Subnetze, in denen Sie Ihre Instances starten möchten.
- Erstellen Sie eine AWS Batch Job-Warteschlange, die Ihrer neuen Rechenumgebung zugeordnet ist. Weitere Informationen finden Sie unter Eine Job-Warteschlange erstellen.
- Erstellen einer Auftragsdefinition für die Ausführung Ihrer Aufträge. Weitere Informationen finden Sie unter Erstellen Sie eine Auftragsdefinition mit einem Knoten .
- Übermitteln eines Auftrags mit Ihrer Auftragsdefinition in die neue Auftragswarteschlange. Dieser Job landet in der Rechenumgebung, die Sie mit Ihrer neuen VPC und Subnetzen erstellt haben. Weitere Informationen finden Sie unter Tutorial: Einen Job einreichen.

Verwenden Sie einen Schnittstellenendpunkt für Access AWS Batch

Sie können AWS PrivateLink damit eine private Verbindung zwischen Ihrer VPC und AWS Batch herstellen. Sie können darauf zugreifen, AWS Batch als ob es in Ihrer VPC wäre, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung zu verwenden. Instances in Ihrer VPC benötigen für den Zugriff AWS Batch keine öffentlichen IP-Adressen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Hierbei handelt es sich um vom

Anforderer verwaltete Netzwerkschnittstellen, die als Eingangspunkt für den Datenverkehr dienen, der für AWS Batch bestimmt ist.

Weitere Informationen finden Sie im Handbuch unter Interface VPC Endpoints.AWS PrivateLink

Überlegungen zu AWS Batch

Bevor Sie einen Schnittstellenendpunkt für einrichten AWS Batch, lesen Sie sich die Eigenschaften und Einschränkungen der Schnittstellenendpunkte im AWS PrivateLink Handbuch durch.

AWS Batch unterstützt Aufrufe aller API-Aktionen über den Schnittstellenendpunkt.

Bevor Sie Schnittstellen-VPC-Endpoints für einrichten AWS Batch, sollten Sie die folgenden Überlegungen beachten:

- Jobs, die den Starttyp Fargate-Ressourcen verwenden, benötigen nicht die Schnittstellen-VPC-Endpunkte für Amazon ECS. Möglicherweise benötigen Sie jedoch Schnittstellen-VPC-Endpunkte für Amazon ECR AWS Batch, Secrets Manager oder Amazon CloudWatch Logs, die in den folgenden Punkten beschrieben werden.
  - Um Jobs auszuführen, müssen Sie die Schnittstellen-VPC-Endpunkte für Amazon ECS erstellen. Weitere Informationen finden Sie unter Interface VPC Endpoints (AWS PrivateLink) im Amazon Elastic Container Service Developer Guide.
  - Damit Ihre Jobs private Images von Amazon ECR abrufen können, müssen Sie die Schnittstellen-VPC-Endpunkte für Amazon ECR erstellen. Weitere Informationen finden Sie unter Schnittstellen-VPC-Endpunkte (AWS PrivateLink) im Amazon Elastic Container-Registry-Benutzerhandbuch.
  - Damit Ihre Jobs sensible Daten aus Secrets Manager abrufen können, müssen Sie die VPC-Schnittstellen-Endpunkte für Secrets Manager erstellen. Weitere Informationen finden Sie unter Verwenden von Secrets Manager mit VPC-Endpunkten im AWS Secrets Manager - Benutzerhandbuch.
  - Wenn Ihre VPC kein Internet-Gateway hat und Ihre Jobs den `awslogs` Protokolltreiber verwenden, um CloudWatch Protokollinformationen an Logs zu senden, müssen Sie einen VPC-Schnittstellen-Endpunkt für CloudWatch Logs erstellen. Weitere Informationen finden Sie unter Using CloudWatch Logs with Interface VPC Endpoints im Amazon CloudWatch Logs-Benutzerhandbuch.
- Jobs, die die EC2 Ressourcen verwenden, erfordern, dass die Container-Instances, auf denen sie gestartet werden, Version `1.25.1` oder höher des Amazon ECS-Container-Agenten ausführen.

Weitere Informationen finden Sie unter Amazon ECS Linux Container Agent-Versionen im Amazon Elastic Container Service Developer Guide.

- VPC-Endpunkte unterstützen derzeit keine regionsübergreifenden Anforderungen. Stellen Sie sicher, dass Sie Ihren Endpunkt innerhalb derselben Region erstellen, in der Sie Ihre API-Aufrufe an AWS Batch ausgeben möchten.
- VPC-Endpunkte unterstützen nur von Amazon bereitgestellten DNS über Amazon Route 53. Wenn Sie Ihre eigene DNS verwenden möchten, können Sie die bedingte DNS-Weiterleitung nutzen. Weitere Informationen finden Sie unter DHCP Options Sets im Amazon VPC-Benutzerhandbuch.
- Die Sicherheitsgruppe für den VPC-Endpunkt müssen eingehende Verbindungen auf Port 443 aus dem privaten Subnetz der VPC zulassen.
- AWS Batch unterstützt im Folgenden keine VPC-Schnittstellenendpunkte: AWS-Regionen
  - Asien-Pazifik (Osaka) (`ap-northeast-3`)
  - Asien-Pazifik (Jakarta) (`ap-southeast-3`)

Erstellen Sie einen Schnittstellenendpunkt für AWS Batch

Sie können einen Schnittstellenendpunkt für die AWS Batch Verwendung entweder der Amazon VPC-Konsole oder der AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für die AWS Batch Verwendung des folgenden Servicenamens:

```
com.amazonaws.region.batch
```

Zum Beispiel:

```
com.amazonaws.us-east-2.batch
```

In der `aws-cn` Partition ist das Format anders:

```
cn.com.amazonaws.region.batch
```

Zum Beispiel:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an die AWS Batch Verwendung des standardmäßigen regionalen DNS-Namens stellen. Beispiel, `batch.us-east-1.amazonaws.com`.

Weitere Informationen finden Sie im AWS PrivateLink Handbuch unter Zugreifen auf einen Dienst über einen Schnittstellenendpunkt.

Erstellen einer Endpunktrichtlinie für Ihren Schnittstellen-Endpunkt

Eine Endpunktrichtlinie ist eine IAM-Ressource, die Sie an einen Schnittstellen-Endpunkt anfügen können. Die standardmäßige Endpunktrichtlinie ermöglicht den vollständigen Zugriff AWS Batch über den Schnittstellenendpunkt. Um den Zugriff zu kontrollieren, der AWS Batch von Ihrer VPC aus gewährt wird, fügen Sie dem Schnittstellenendpunkt eine benutzerdefinierte Endpunktrichtlinie hinzu.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Principals, die Aktionen ausführen können (AWS-Konten, Benutzer und IAM-Rollen).
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuern des Zugriffs auf Services mit Endpunktrichtlinien im AWS PrivateLink -Leitfaden.

Beispiel: VPC-Endpunktrichtlinie für Aktionen AWS Batch

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Endpunktrichtlinie. Wenn Sie diese Richtlinie an Ihren Schnittstellenendpunkt anhängen, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten AWS Batch Aktionen.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "batch:SubmitJob",
            "batch:ListJobs",
            "batch:DescribeJobs"
         ],
         "Resource":"*"
```

```
        }
    ]
}
```

Konformitätsprüfung für AWS Batch

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- 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.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um

Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.

- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Sicherheit der Infrastruktur in AWS Batch

Als verwalteter Dienst AWS Batch ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff AWS Batch 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.

Sie können diese API-Operationen von jedem Netzwerkstandort aus aufrufen, AWS Batch unterstützt jedoch ressourcenbasierte Zugriffsrichtlinien, die Einschränkungen auf der Grundlage der Quell-IP-Adresse beinhalten können. Sie können auch AWS Batch Richtlinien verwenden, um den Zugriff

von bestimmten Amazon Virtual Private Cloud (Amazon VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf eine bestimmte AWS Batch Ressource effektiv nur von der spezifischen VPC innerhalb des AWS Netzwerks isoliert.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine Entität, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine Entität mit größeren Rechten zwingen kann, die Aktion auszuführen. In der AWS Tat kann ein dienstübergreifender Identitätswechsel zu einem Problem mit dem verwirrten Stellvertreter führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die der AWS Batch Ressource einen anderen Dienst gewähren. Wenn der `aws:SourceArn`-Wert die Konto-ID nicht enthält, z. B. einen Amazon-S3-Bucket-ARN, müssen Sie beide globale Bedingungskontextschlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Der Wert von `aws:SourceArn` muss die Ressource sein, die AWS Batch gespeichert wird.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Kontextbedingungsschlüssel `aws:SourceArn` mit Platzhalterzeichen (`*`) für die unbekannten Teile des ARN. Beispiel, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Die folgenden Beispiele zeigen, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel verwenden können, AWS Batch um das Problem des verwirrten Stellvertreters zu vermeiden.

Beispiel: Rolle für den Zugriff auf nur eine Rechenumgebung

Die folgende Rolle kann nur für den Zugriff auf eine Rechenumgebung verwendet werden. Der Jobname muss als angegeben werden*, da die Job-Warteschlange mehreren Rechenumgebungen zugeordnet werden kann.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Beispiel: Rolle für den Zugriff auf mehrere Rechenumgebungen

Die folgende Rolle kann für den Zugriff auf mehrere Rechenumgebungen verwendet werden. Der Jobname muss angegeben werden, * da die Job-Warteschlange mehreren Rechenumgebungen zugeordnet werden kann.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

AWS Batch API-Aufrufe protokollieren mit AWS CloudTrail

AWS Batch ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden AWS Batch. CloudTrail erfasst alle API-Aufrufe AWS Batch als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS Batch Konsole und Codeaufrufen für die AWS Batch API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von

CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS Batch. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde AWS Batch, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

Themen

- AWS Batch Informationen in CloudTrail
- Referenz: Grundlegendes zu AWS Batch Protokolldateieinträgen

AWS Batch Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn eine Aktivität in stattfindet AWS Batch, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für AWS Batch, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. 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
- Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle AWS Batch Aktionen werden von der https://docs.aws.amazon.com/batch/ neuesten APIReference Version protokolliert// CloudTrail und sind dort dokumentiert. Zum Beispiel werden durch Aufrufe der `SubmitJob`-, `ListJobs`- und `DescribeJobs`-Abschnitte Einträge in den CloudTrail -Protokolldateien generiert.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anforderung mit Root- oder IAM-Benutzer-Anmeldeinformationen ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen Dienst gestellt wurde. AWS

Weitere Informationen finden Sie unter CloudTrail userIdentity-Element.

Referenz: Grundlegendes zu AWS Batch Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis ist eine einzelne Anforderung aus einer beliebigen Quelle und enthält Informationen zur angeforderten Aktion, zu Datum und Uhrzeit der Aktion, zu den Anforderungsparametern usw. CloudTrail-Protokolldateien stellen kein geordnetes Stack-Trace der öffentlichen API-Aufrufe dar. Daher werden sie nicht in einer bestimmten Reihenfolge angezeigt.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateComputeEnvironment` Aktion demonstriert.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
```

```
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2017-12-20T00:48:46Z",
  "eventSource": "batch.amazonaws.com",
  "eventName": "CreateComputeEnvironment",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.1",
  "userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
  "requestParameters": {
    "computeResources": {
      "subnets": [
        "subnet-5eda8e04"
      ],
      "tags": {
        "testBatchTags": "CLI testing CE"
      },
      "desiredvCpus": 0,
      "minvCpus": 0,
      "instanceTypes": [
        "optimal"
      ],
      "securityGroupIds": [
        "sg-aba9e8db"
      ],
      "instanceRole": "ecsInstanceRole",
      "maxvCpus": 128,
      "type": "EC2"
    },
    "state": "ENABLED",
    "type": "MANAGED",
    "computeEnvironmentName": "Test"
  },
  "responseElements": {
    "computeEnvironmentName": "Test",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/
Test"
  },
```

```
  "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
  "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "012345678910"
}
```

Problembehandlung bei AWS Batch IAM

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS Batch IAM auftreten können.

Themen

- Ich bin nicht autorisiert, eine Aktion in AWS Batch auszuführen.
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS Batch Ressourcen ermöglichen

Ich bin nicht autorisiert, eine Aktion in AWS Batch auszuführen.

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion auszuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven *`my-example-widget`*-Ressource zu verwenden, jedoch nicht über `batch:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 batch:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *`my-example-widget`* auf die Ressource `batch:`*`GetWidget`* zugreifen zu können. Weitere Informationen zum Erteilen von Berechtigungen zur Weitergabe einer Rolle finden Sie unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS Dienst.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS Batchübergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in AWS Batch auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS Batch Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen AWS Batch unterstützt werden, finden Sie unter. Wie AWS Batch funktioniert mit IAM

- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

AWS Step Functions

Sie können die AWS Batch Konsole verwenden, um Details zu Ihren Step Functions-Zustandsmaschinen und den von ihnen verwendeten Funktionen anzuzeigen.

Sections

- Tutorial: Details zur Zustandsmaschine anzeigen
- Tutorial: Bearbeiten Sie eine Zustandsmaschine
- Tutorial: Führen Sie eine Zustandsmaschine aus

Tutorial: Details zur Zustandsmaschine anzeigen

In der AWS Batch Konsole wird eine Liste Ihrer aktuellen Zustandsmaschinen angezeigt AWS-Region , die mindestens einen Workflow-Schritt enthalten, mit dem ein AWS Batch Job gesendet wird.

Wählen Sie einen Zustandsautomaten aus, um eine grafische Darstellung des Workflows anzuzeigen. Blau hervorgehobene Schritte stehen für AWS Batch Jobs. Verwenden Sie die Diagrammsteuerelemente, um das Diagramm zu vergrößern, zu verkleinern und zu zentrieren.

> **Note**
>
> Wenn JsonPath in der State-Machine-Definition dynamisch auf einen AWS Batch Job verwiesen wird, können die Funktionsdetails nicht in der AWS Batch Konsole angezeigt werden. Stattdessen wird der Jobname als dynamische Referenz aufgeführt und die entsprechenden Schritte im Diagramm sind ausgegraut.

So lassen Sie sich Details zu den Zustandsautomaten anzeigen

1. Öffnen Sie die AWS Batch Konsolenseite Workflow-Orchestrierung powered by Step Functions.
2. Wählen Sie einen Zustandsautomaten aus.
   <result>

   Die AWS Batch Konsole öffnet die Seite „Details“.
   </result>

Weitere Informationen finden Sie unter Step Functions im AWS Step Functions -Entwicklerhandbuch.

Tutorial: Bearbeiten Sie eine Zustandsmaschine

Wenn Sie eine Zustandsmaschine bearbeiten möchten, AWS Batch wird die Seite Definition bearbeiten der Step Functions Functions-Konsole geöffnet.

So bearbeiten Sie einen Zustandsautomaten

1. Öffnen Sie die AWS Batch Konsolenseite Workflow-Orchestrierung powered by Step Functions.
2. Wählen Sie einen Zustandsautomaten aus.
3. Wählen Sie Edit (Bearbeiten) aus.

   In der Step-Functions-Konsole wird die Seite Definition bearbeiten geöffnet.
4. Bearbeiten Sie den Zustandsautomaten und wählen Sie Speichern.

Weitere Informationen zum Bearbeiten von Zustandsautomaten finden Sie unter Step-Functions-Zustandsautomaten-Sprache im AWS Step Functions -Entwicklerhandbuch.

Tutorial: Führen Sie eine Zustandsmaschine aus

Wenn Sie eine Zustandsmaschine ausführen möchten, AWS Batch wird die Seite Neue Ausführung der Step Functions-Konsole geöffnet.

So führen Sie einen Zustandsautomaten aus

1. Öffnen Sie die AWS Batch Konsolenseite Workflow-Orchestrierung powered by Step Functions.
2. Wählen Sie einen Zustandsautomaten aus.
3. Wählen Sie Execute (Ausführen).

   In der Step-Functions-Konsole wird die Seite Neue Ausführung geöffnet.
4. (Optional) Bearbeiten Sie den Zustandsautomaten und wählen Sie Start execution (Ausführung starten) aus.

Weitere Informationen zum Ausführen von Zustandsautomaten finden Sie unter Ausführungskonzepte für Step-Functions-Zustandsautomaten im AWS Step Functions -Entwicklerhandbuch.

AWS Batch Event-Stream für Amazon EventBridge

Sie können den AWS Batch Event-Stream für Amazon verwenden EventBridge , um nahezu in Echtzeit Benachrichtigungen über den aktuellen Status der Jobs in Ihren Job-Warteschlangen zu erhalten.

Sie können ihn nutzen EventBridge , um weitere Einblicke in Ihren AWS Batch Service zu erhalten. Insbesondere können Sie damit den Fortschritt von Jobs überprüfen, AWS Batch benutzerdefinierte Workflows erstellen, Nutzungsberichte oder Metriken erstellen oder Ihre eigenen Dashboards erstellen. Mit AWS Batch und EventBridge benötigen Sie keinen Planungs- und Überwachungscode, der kontinuierlich AWS Batch nach Änderungen des Auftragsstatus abfragt. Stattdessen können Sie Änderungen des AWS Batch Jobstatus mithilfe einer Vielzahl von EventBridge Amazon-Zielen asynchron verarbeiten. Dazu gehören AWS Lambda Amazon Simple Queue Service, Amazon Simple Notification Service oder Amazon Kinesis Data Streams.

Es wird sichergestellt, dass Ereignisse aus dem AWS Batch Event-Stream mindestens einmal zugestellt werden. Falls doppelte Ereignisse übermittelt werden, enthält das Ereignis ausreichend Informationen, um Duplikate zu erkennen. Auf diese Weise können Sie den Zeitstempel des Ereignisses und den Jobstatus vergleichen.

AWS Batch Jobs sind als EventBridge Ziele verfügbar. Mithilfe einfacher Regeln können Sie Ereignisse zuordnen und als Reaktion darauf AWS Batch Jobs einreichen. Weitere Informationen finden Sie unter Was ist EventBridge? im EventBridge Amazon-Benutzerhandbuch. Sie können es auch verwenden EventBridge , um automatisierte Aktionen zu planen, die sich zu bestimmten Zeiten selbst auslösen, indem Sie Ausdrücke verwenden cron oder bewerten. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen einer EventBridge Amazon-Regel, die nach einem Zeitplan ausgeführt wird. Ein Beispiel-Walkthrough finden Sie unter AWS Batch Jobs als EventBridge Ziele. Informationen zur Verwendung des EventBridge Schedulers finden Sie unter Amazon EventBridge Scheduler einrichten im EventBridge Amazon-Benutzerhandbuch.

Themen

- AWS Batch Ereignisse
- Tutorial: AWS Benutzerbenachrichtigungen verwenden mit AWS Batch
- AWS Batch Jobs als EventBridge Ziele
- Tutorial: Halten Sie Ausschau nach AWS Batch Job-Events mit EventBridge
- Tutorial: Senden von Amazon Simple Notification Service-Benachrichtigungen für fehlgeschlagene Auftragsereignisse

AWS Batch Ereignisse

AWS Batch sendet Ereignisse zur Änderung des Jobstatus an EventBridge. AWS Batch verfolgt den Status Ihrer Jobs. Wenn sich der Status eines zuvor eingereichten Jobs ändert, wird ein Ereignis ausgelöst. Zum Beispiel, wenn ein Job im `RUNNING` Status in den Status wechselt. `FAILED` Diese Ereignisse sind als Auftragsstatus-Änderungsereignisse klassifiziert.

> **Note**
>
> AWS Batch könnte in future weitere Ereignistypen, Quellen und Details hinzufügen. Wenn Sie JSON-Ereignisdaten programmgesteuert deserialisieren, stellen Sie sicher, dass Ihre Anwendung darauf vorbereitet ist, unbekannte Eigenschaften zu verarbeiten. Dies dient dazu, Probleme zu vermeiden, falls und wann diese zusätzlichen Eigenschaften hinzugefügt werden.

Themen

- Ressource: Ereignisse zur Änderung des Berufsstatus
- Ressource: Blockierte Ereignisse in der Jobwarteschlange

Ressource: Ereignisse zur Änderung des Berufsstatus

Jedes Mal, wenn sich der Status eines vorhandenen (zuvor eingereichten) Jobs ändert, wird ein Ereignis erstellt. Weitere Informationen zu AWS Batch Jobstatus finden Sie unterAuftragsstatus.

> **Note**
>
> Für die erste Stelleneingabe werden keine Ereignisse erstellt.

Example Auftragsstatus-Änderungsereignis

Ereignisse zur Änderung des Jobstatus werden im folgenden Format übermittelt. Der `detail` Abschnitt ähnelt dem JobDetailObjekt, das von einer DescribeJobsAPI-Operation in der AWS Batch API-Referenz zurückgegeben wurde. Weitere Informationen zu EventBridge Parametern finden Sie unter Ereignisse und Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

```
{
```

```
    "version": "0",
    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job State Change",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    ],
    "detail": {
        "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "jobName": "event-test",
        "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
        "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
        "status": "RUNNABLE",
        "attempts": [],
        "createdAt": 1641944200058,
        "retryStrategy": {
            "attempts": 2,
            "evaluateOnExit": []
        },
        "dependsOn": [],
        "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
        "parameters": {},
        "container": {
            "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
            "command": [
                "sleep",
                "600"
            ],
            "volumes": [],
            "environment": [],
            "mountPoints": [],
            "ulimits": [],
            "networkInterfaces": [],
            "resourceRequirements": [
                {
                    "value": "2",
                    "type": "VCPU"
                }, {
```

```
                    "value": "256",
                    "type": "MEMORY"
                }
            ],
            "secrets": []
        },
        "tags": {
            "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
        },
        "propagateTags": false,
        "platformCapabilities": []
    }
}
```

Ressource: Blockierte Ereignisse in der Jobwarteschlange

Jedes Mal, wenn ein Job im `RUNNABLE` Status AWS Batch erkannt und somit eine Warteschlange blockiert wird, wird ein Ereignis in Amazon CloudWatch Events erstellt. Weitere Informationen zu den unterstützten Ursachen für blockierte Warteschlangen finden Sie unterJobs, die in einem `RUNNABLE` Status feststecken. Derselbe Grund ist auch in dem `statusReason` Feld in der `DescribeJobs`API-Aktion verfügbar.

Example Auftragsstatus-Änderungsereignis

Ereignisse zur Änderung des Jobstatus werden im folgenden Format übermittelt. Der `detail` Abschnitt ähnelt dem JobDetailObjekt, das von einer DescribeJobsAPI-Operation in der AWS Batch API-Referenz zurückgegeben wurde. Weitere Informationen zu EventBridge Parametern finden Sie unter Ereignisse und Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

```
{
    "version": "0",
    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job Queue Blocked",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
```

```
    ],
    "detail": {
        "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "jobName": "event-test",
        "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
        "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
        "status": "RUNNABLE",
        "statusReason": "blocked-reason",
        "attempts": [],
        "createdAt": 1641944200058,
        "retryStrategy": {
            "attempts": 2,
            "evaluateOnExit": []
        },
        "dependsOn": [],
        "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
        "parameters": {},
        "container": {
            "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
            "command": [
                "sleep",
                "600"
            ],
            "volumes": [],
            "environment": [],
            "mountPoints": [],
            "ulimits": [],
            "networkInterfaces": [],
            "resourceRequirements": [
                {
                    "value": "2",
                    "type": "VCPU"
                }, {
                    "value": "256",
                    "type": "MEMORY"
                }
            ],
            "secrets": []
        },
        "tags": {
```

```
            "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
        },
        "propagateTags": false,
        "platformCapabilities": []
    }
}
```

Tutorial: AWS Benutzerbenachrichtigungen verwenden mit AWS Batch

Sie können AWS Benutzerbenachrichtigungen verwenden, um Zustellungskanäle einzurichten, um über AWS Batch Ereignisse informiert zu werden. Sie erhalten eine Benachrichtigung, wenn ein Ereignis einer von Ihnen angegebenen Regel entspricht. Sie können Benachrichtigungen für Ereignisse über mehrere Kanäle erhalten, darunter E-Mail, Amazon Q Developer in Chat-Anwendungen, Chat-Benachrichtigungen oder AWS Console Mobile ApplicationPush-Benachrichtigungen. Benachrichtigungen werden auch im Console Notifications Center angezeigt. Die Funktion für Benutzerbenachrichtigungen unterstützt eine Aggregation, wodurch die Anzahl der Benachrichtigungen, die Sie bei bestimmten Ereignissen erhalten, reduziert werden kann.

So konfigurieren Sie Benutzerbenachrichtigungen in AWS Batch:

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie Dashboard.
3. Wählen Sie Benachrichtigungen konfigurieren.
4. Wählen Sie unter AWS Benutzerbenachrichtigungen die Option Benachrichtigungskonfiguration erstellen aus.

Weitere Informationen zum Konfigurieren und Anzeigen von Benutzerbenachrichtigungen finden Sie unter Erste Schritte mit AWS Benutzerbenachrichtigungen.

AWS Batch Jobs als EventBridge Ziele

Amazon EventBridge liefert nahezu in Echtzeit einen Stream von Systemereignissen, die Änderungen an den Ressourcen von Amazon Web Services beschreiben. In der Regel sind Jobs AWS Batch auf Amazon Elastic Container Service, Amazon Elastic Kubernetes Service und AWS Fargate als Ziele

verfügbar. EventBridge Mithilfe einfacher Regeln können Sie Ereignisse zuordnen und als Reaktion darauf AWS Batch Jobs einreichen. Weitere Informationen finden Sie unter Was ist EventBridge? im EventBridge Amazon-Benutzerhandbuch.

Sie können es auch verwenden EventBridge , um automatisierte Aktionen zu planen, die zu bestimmten Zeiten aufgerufen werden, indem Sie Ausdrücke verwenden cron oder bewerten. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen einer EventBridge Amazon-Regel, die nach einem Zeitplan ausgeführt wird.

Informationen zum Erstellen einer Regel, die ausgeführt wird, wenn ein Ereignis mit einem Ereignismuster übereinstimmt, finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.

Zu den häufigsten Anwendungsfällen für AWS Batch Jobs als EventBridge Ziel gehören die folgenden Anwendungsfälle:

- Ein geplanter Job wird in regelmäßigen Zeitintervallen ausgeführt. Ein cron Job wird beispielsweise nur zu Zeiten mit geringer Auslastung ausgeführt, wenn Amazon EC2 Spot-Instances günstiger sind.
- Ein AWS Batch Job wird als Reaktion auf einen API-Vorgang ausgeführt, der angemeldet CloudTrail ist. Beispielsweise wird ein Job immer dann übermittelt, wenn ein Objekt in einen bestimmten Amazon S3 S3-Bucket hochgeladen wird. Jedes Mal, wenn dies passiert, übergibt der EventBridge Eingangstransformator den Bucket und den Schlüsselnamen des Objekts an AWS Batch Parameter.

> **Note**
>
> In diesem Szenario müssen sich alle zugehörigen AWS Ressourcen in derselben Region befinden. Dazu gehören Ressourcen wie der Amazon S3 S3-Bucket, die EventBridge Regel und CloudTrail Protokolle.

Bevor Sie AWS Batch Jobs mit EventBridge Regeln und Zielen einreichen können, benötigt der EventBridge Service mehrere Berechtigungen, um AWS Batch Jobs auszuführen. Wenn Sie in der EventBridge Konsole eine Regel erstellen, die einen AWS Batch Job als Ziel angibt, können Sie auch diese Rolle erstellen. Weitere Informationen über die erforderlichen Service-Prinzipal- und IAM;-Berechtigungen für dies Rolle finden Sie unter EventBridge IAM-Rolle.

Themen

- Tutorial: Einen geplanten AWS Batch Job erstellen
- Tutorial: Erstellen Sie eine Regel mit einem Ereignismuster
- Tutorial: Übergeben Sie mithilfe des EventBridge Eingangstransformators Ereignisinformationen an ein AWS Batch Ziel nach einem Zeitplan

Tutorial: Einen geplanten AWS Batch Job erstellen

Das folgende Verfahren beschreibt, wie Sie einen geplanten AWS Batch Job und die erforderliche EventBridge IAM-Rolle erstellen.

Um einen geplanten AWS Batch Job zu erstellen mit EventBridge

> **Note**
>
> Dieses Verfahren funktioniert für alle Jobs AWS Batch auf Amazon ECS, Amazon EKS und AWS Fargate.

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Regeln aus.
4. Wählen Sie Regel erstellen aus.
5. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung ein. Der Name kann bis zu 64 Zeichen umfassen. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

   > **Note**
   >
   > Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.

6. (Optional) Geben Sie unter Beschreibung eine Beschreibung für die Regel ein.
7. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel mit Ereignissen aus Ihrem eigenen Konto übereinstimmt, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.

8. (Optional) Schalten Sie die Regel für den ausgewählten Bus aus, wenn Sie die Regel nicht sofort ausführen möchten.
9. Wählen Sie unter Rule type (Regeltyp) die Option Schedule (Zeitplan) aus.
10. Wählen Sie Weiter, um die Regel zu erstellen, oder Weiter.
11. Gehen Sie bei Schedule pattern (Zeitplanmuster) wie folgt vor:
    - Wählen Sie Ein detaillierter Zeitplan, der zu einer bestimmten Zeit ausgeführt wird, z. B. um 8:00 Uhr aus. PST am ersten Montag eines jeden Monats und geben Sie dann einen Cron-Ausdruck ein. Weitere Informationen finden Sie unter Cron Expressions im EventBridge Amazon-Benutzerhandbuch.
    - Wählen Sie einen Zeitplan, der regelmäßig ausgeführt wird, z. B. alle 10 Minuten. und geben Sie dann einen Preisausdruck ein.
12. Wählen Sie Weiter.
13. Für Target types (Zieltypen), wählen Sie AWS-Service aus.
14. Wählen Sie für Wählen Sie ein Ziel die Option Batch-Job-Warteschlange aus. Konfigurieren Sie dann Folgendes:
    - Job queue (Auftragswarteschlange): Geben Sie den Amazon-Ressourcennamen (ARN) der Auftragswarteschlange ein, in der Ihr Auftrag eingeplant werden soll.
    - Auftragsdefinition: Geben Sie den Namen und die Revision oder den vollständigen ARN der Auftragsdefinition ein, die für Ihren Auftrag verwendet werden sollen.
    - Job name (Auftragsname): Geben Sie einen Namen für Ihren Auftrag ein.
    - Array-Größe: (Optional) Geben Sie eine Array-Größe für Ihren Auftrag ein, um mehrere Kopien davon auszuführen. Weitere Informationen finden Sie unter Ordnen Sie Jobs an.
    - Auftragsversuche: (Optional) Geben Sie an, wie oft die Ausführung Ihres Auftrags versucht werden soll, bevor festgelegt wird, dass er fehlgeschlagen ist. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.
15. Für Zieltypen von Batch-Job-Warteschlangen ist die Erlaubnis EventBridge erforderlich, Ereignisse an das Ziel zu senden. EventBridge kann die IAM-Rolle erstellen, die für die Ausführung Ihrer Regel erforderlich ist. Führen Sie eine der folgenden Aktionen aus:
    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Eine neue Rolle für diese spezifische Ressource erstellen.
    - Um eine IAM-Rolle zu verwenden, die Sie bereits erstellt haben, wählen Sie Bestehende Rolle verwenden.

16. (Optional) Erweitern Sie Additional settings (Zusätzliche Einstellungen).
    a. Wählen Sie unter Zieleingabe konfigurieren aus, wie der Text eines Ereignisses verarbeitet wird, bevor er an das Ziel übergeben wird.
    b. Geben Sie unter Maximales Alter des Ereignisses das Zeitintervall an, in dem unverarbeitete Ereignisse aufbewahrt werden.
    c. Geben Sie unter Wiederholungsversuche ein, wie oft ein Ereignis wiederholt werden soll.
    d. Wählen Sie für die Warteschlange „Unzustellbarer Briefe“ eine Option für die Behandlung unbearbeiteter Ereignisse aus. Geben Sie bei Bedarf die Amazon SQS SQS-Warteschlange an, die als Warteschlange für unzustellbare Briefe verwendet werden soll.
17. (Optional) Wählen Sie Add another target (Weiteres Ziel hinzufügen) aus, um ein weiteres Ziel für diese Regel hinzuzufügen.
18. Wählen Sie Weiter.
19. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus, um eine Ressourcenbezeichnung für die Regel hinzuzufügen. Weitere Informationen finden Sie unter EventBridge Amazon-Tags.
20. Wählen Sie Weiter.
21. Überprüfen und erstellen finden Sie in den Konfigurationsschritten. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Regel erstellen.

Weitere Informationen zum Erstellen von Regeln finden Sie unter Erstellen einer EventBridge Amazon-Regel, die nach einem Zeitplan ausgeführt wird im EventBridge Amazon-Benutzerhandbuch.

Tutorial: Erstellen Sie eine Regel mit einem Ereignismuster

Das folgende Verfahren beschreibt, wie Sie eine Regel mit einem Ereignismuster erstellen.

Um eine Regel zu erstellen, die das Ereignis an ein Ziel sendet, wenn das Ereignis einem definierten Muster entspricht

> **Note**
>
> Dieses Verfahren funktioniert für alle Jobs AWS Batch auf Amazon ECS, Amazon EKS und AWS Fargate.

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Regeln aus.
4. Wählen Sie Regel erstellen aus.
5. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung ein. Der Name kann bis zu 64 Zeichen umfassen. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

   > **Note**
   >
   > Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.

6. (Optional) Geben Sie unter Beschreibung eine Beschreibung für die Regel ein.
7. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel mit Ereignissen aus Ihrem eigenen Konto übereinstimmt, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
8. (Optional) Schalten Sie die Regel für den ausgewählten Bus aus, wenn Sie die Regel nicht sofort ausführen möchten.
9. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
10. Wählen Sie Weiter.
11. Wählen Sie als Ereignisquelle AWS Ereignis oder EventBridge Partnerereignisse aus.
12. (Optional) Für ein Beispielereignis:
    a. Wählen Sie für Beispiel-Ereignistyp die Option AWS Ereignisse aus.
    b. Wählen Sie für Beispielereignisse die Option Batch Job State Change aus.
13. Wählen Sie für Erstellungsmethode die Option Musterformular verwenden aus.
14. Für Event-Pattern:
    a. Wählen Sie für Ereignisquelle die Option AWS-Services aus.
    b. Wählen Sie für AWS-ServiceBatch.
    c. Wählen Sie als Ereignistyp die Option Batch Job State Change aus.
15. Wählen Sie Weiter.

16. Für Target types (Zieltypen), wählen Sie AWS-Service aus.
17. Wählen Sie unter Ziel auswählen einen Zieltyp aus. Wählen Sie beispielsweise Batch-Job-Warteschlange. Geben Sie dann Folgendes an:
    - Job queue (Auftragswarteschlange): Geben Sie den Amazon-Ressourcennamen (ARN) der Auftragswarteschlange ein, in der Ihr Auftrag eingeplant werden soll.
    - Auftragsdefinition: Geben Sie den Namen und die Revision oder den vollständigen ARN der Auftragsdefinition ein, die für Ihren Auftrag verwendet werden sollen.
    - Job name (Auftragsname): Geben Sie einen Namen für Ihren Auftrag ein.
    - Array-Größe: (Optional) Geben Sie eine Array-Größe für Ihren Auftrag ein, um mehrere Kopien davon auszuführen. Weitere Informationen finden Sie unter Ordnen Sie Jobs an.
    - Auftragsversuche: (Optional) Geben Sie an, wie oft die Ausführung Ihres Auftrags versucht werden soll, bevor festgelegt wird, dass er fehlgeschlagen ist. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.
18. Für Zieltypen von Batch-Job-Warteschlangen ist die Erlaubnis EventBridge erforderlich, Ereignisse an das Ziel zu senden. EventBridge kann die IAM-Rolle erstellen, die für die Ausführung Ihrer Regel erforderlich ist. Führen Sie eine der folgenden Aktionen aus:
    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Create a new role for this specific resource (Eine neue Rolle für diese spezifische Ressource erstellen).
    - Wenn Sie eine zuvor erstellte IAM-Rolle verwenden möchten, wählen Sie Use existing role (Vorhandene Rolle verwenden).
19. (Optional) Erweitern Sie Additional settings (Zusätzliche Einstellungen).
    a. Wählen Sie unter Zieleingabe konfigurieren aus, wie Text aus einem Ereignis verarbeitet wird.
    b. Geben Sie für Maximales Alter des Ereignisses das Zeitintervall an, in dem unverarbeitete Ereignisse aufbewahrt werden.
    c. Geben Sie unter Wiederholungsversuche ein, wie oft ein Ereignis wiederholt werden soll.
    d. Wählen Sie für die Warteschlange „Unzustellbarer Briefe“ eine Option für die Behandlung unbearbeiteter Ereignisse aus. Geben Sie bei Bedarf die Amazon SQS SQS-Warteschlange an, die als Warteschlange für unzustellbare Briefe verwendet werden soll.
20. (Optional) Wählen Sie Weiteres Ziel hinzufügen, um ein zusätzliches Ziel hinzuzufügen.
21. Wählen Sie Weiter.

22. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus, um ein Ressourcenlabel hinzuzufügen. Weitere Informationen finden Sie unter EventBridge Amazon-Tags im EventBridge Amazon-Benutzerhandbuch.
23. Wählen Sie Weiter.
24. Überprüfen und erstellen finden Sie in den Konfigurationsschritten. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Regel erstellen aus.

    Weitere Informationen zum Erstellen von Regeln finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erstellen von EventBridge Amazon-Regeln, die auf Ereignisse reagieren.

Tutorial: Übergeben Sie mithilfe des EventBridge Eingangstransformators Ereignisinformationen an ein AWS Batch Ziel nach einem Zeitplan

Sie können den EventBridge Eingangstransformator verwenden, um Ereignisinformationen AWS Batch in einer Job-Einreichung weiterzuleiten. Dies kann besonders nützlich sein, wenn Sie Jobs aufgrund anderer AWS Ereignisinformationen aufrufen. Ein Beispiel ist ein Objekt-Upload in einen Amazon S3 S3-Bucket. Sie können auch eine Auftragsdefinition mit Parameterersatzwerten im Befehl des Containers verwenden. Der EventBridge Eingangstransformator kann die Parameterwerte auf der Grundlage der Ereignisdaten bereitstellen.

Anschließend erstellen Sie ein AWS Batch Ereigniszieł, das Informationen aus dem Ereignis, das es auslöst, analysiert und in ein `parameters` Objekt umwandelt. Wenn der Job ausgeführt wird, werden die Parameter des Trigger-Ereignisses an den Befehl des Job-Containers übergeben.

> **Note**
>
> In diesem Szenario müssen sich alle AWS Ressourcen (wie Amazon S3 S3-Buckets, EventBridge Regeln und CloudTrail Protokolle) in derselben Region befinden.

Um ein AWS Batch Ziel zu erstellen, das den Eingangstransformator verwendet

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Regeln aus.

4. Wählen Sie Regel erstellen aus.
5. Geben Sie unter Name einen eindeutigen Namen für Ihre Rechenumgebung ein. Der Name kann bis zu 64 Zeichen umfassen. Er kann Groß- und Kleinbuchstaben, Zahlen, Bindestriche (-) und Unterstriche (_) enthalten.

> **Note**
>
> Eine Regel kann nicht denselben Namen wie eine andere Regel im selben AWS-Region und im selben Event-Bus haben.

6. (Optional) Geben Sie unter Beschreibung eine Beschreibung für die Regel ein.
7. Wählen Sie für Event Bus den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel mit Ereignissen aus Ihrem eigenen Konto übereinstimmt, wählen Sie Standard aus. Wenn ein AWS-Service in Ihrem Konto ein Ereignis ausgibt, wird es immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
8. (Optional) Schalten Sie die Regel für den ausgewählten Bus aus, wenn Sie die Regel nicht sofort ausführen möchten.
9. Wählen Sie unter Rule type (Regeltyp) die Option Schedule (Zeitplan) aus.
10. Wählen Sie Weiter, um die Regel zu erstellen, oder Weiter.
11. Gehen Sie bei Schedule pattern (Zeitplanmuster) wie folgt vor:
    - Wählen Sie Ein detaillierter Zeitplan, der zu einer bestimmten Zeit ausgeführt wird, z. B. um 8:00 Uhr aus. PST am ersten Montag eines jeden Monats und geben Sie dann einen Cron-Ausdruck ein. Weitere Informationen finden Sie unter Cron Expressions im EventBridge Amazon-Benutzerhandbuch.
    - Wählen Sie einen Zeitplan, der regelmäßig ausgeführt wird, z. B. alle 10 Minuten. und geben Sie dann einen Preisausdruck ein.
12. Wählen Sie Weiter.
13. Für Target types (Zieltypen), wählen Sie AWS-Service aus.
14. Wählen Sie für Wählen Sie ein Ziel die Option Batch-Job-Warteschlange aus. Konfigurieren Sie dann Folgendes:
    - Job queue (Auftragswarteschlange): Geben Sie den Amazon-Ressourcennamen (ARN) der Auftragswarteschlange ein, in der Ihr Auftrag eingeplant werden soll.

- Auftragsdefinition: Geben Sie den Namen und die Revision oder den vollständigen ARN der Auftragsdefinition ein, die für Ihren Auftrag verwendet werden sollen.
- Job name (Auftragsname): Geben Sie einen Namen für Ihren Auftrag ein.
- Array-Größe: (Optional) Geben Sie eine Array-Größe für Ihren Auftrag ein, um mehrere Kopien davon auszuführen. Weitere Informationen finden Sie unter Ordnen Sie Jobs an.
- Auftragsversuche: (Optional) Geben Sie an, wie oft die Ausführung Ihres Auftrags versucht werden soll, bevor festgelegt wird, dass er fehlgeschlagen ist. Weitere Informationen finden Sie unter Automatisierte Auftragswiederholungen.

15. Für Zieltypen von Batch-Job-Warteschlangen ist die Erlaubnis EventBridge erforderlich, Ereignisse an das Ziel zu senden. EventBridge kann die IAM-Rolle erstellen, die für die Ausführung Ihrer Regel erforderlich ist. Führen Sie eine der folgenden Aktionen aus:
    - Um automatisch eine IAM-Rolle zu erstellen, wählen Sie Eine neue Rolle für diese spezifische Ressource erstellen.
    - Um eine IAM-Rolle zu verwenden, die Sie bereits erstellt haben, wählen Sie Bestehende Rolle verwenden.
16. (Optional) Erweitern Sie Additional settings (Zusätzliche Einstellungen).
17. Wählen Sie im Abschnitt Additional settings (Zusätzliche Einstellungen) unter Configure target input (Zieleingabe konfigurieren) die Option Input Transformer (Eingabetransformator).
18. Wählen Sie Configure input transformer (Eingabetransformator konfigurieren).
19. (Optional) Für ein Beispielereignis:
    a. Wählen Sie für Beispiel-Ereignistyp die Option AWS Ereignisse aus.
    b. Wählen Sie für Beispielereignisse die Option Batch Job State Change aus.
20. Geben Sie im Abschnitt Target input transformer (Zieleingabetransformator) für Input path (Eingabepfad) die Werte an, die aus dem auslösenden Ereignis analysiert werden sollen. Um beispielsweise das Batch Job State Change-Ereignis zu analysieren, verwenden Sie das folgende JSON-Format.

```
{
    "instance": "$.detail.jobId",
    "state": "$.detail.status"
}
```

21. Geben Sie für Vorlage Folgendes ein.

```
{
    "instance": <jobId> ,
    "status": <status>
}
```

22. Wählen Sie Bestätigen aus.
23. Geben Sie unter Maximales Alter des Ereignisses das Zeitintervall an, in dem unverarbeitete Ereignisse aufbewahrt werden.
24. Geben Sie unter Wiederholungsversuche ein, wie oft ein Ereignis wiederholt werden soll.
25. Wählen Sie für die Warteschlange „Unzustellbarer Briefe“ eine Option für die Behandlung unbearbeiteter Ereignisse aus. Geben Sie bei Bedarf die Amazon SQS SQS-Warteschlange an, die als Warteschlange für unzustellbare Briefe verwendet werden soll.
26. (Optional) Wählen Sie Weiteres Ziel hinzufügen, um ein zusätzliches Ziel hinzuzufügen.
27. Wählen Sie Weiter.
28. (Optional) Wählen Sie für Tags die Option Neues Tag hinzufügen aus, um ein Ressourcenlabel hinzuzufügen. Weitere Informationen finden Sie unter EventBridge Amazon-Tags im EventBridge Amazon-Benutzerhandbuch.
29. Wählen Sie Weiter.
30. Überprüfen und erstellen finden Sie in den Konfigurationsschritten. Wenn Sie Änderungen vornehmen müssen, wählen Sie Edit (Bearbeiten). Wenn Sie fertig sind, wählen Sie Regel erstellen aus.

Tutorial: Halten Sie Ausschau nach AWS Batch Job-Events mit EventBridge

In diesem Tutorial richten Sie eine einfache AWS Lambda Funktion ein, die auf AWS Batch Job-Ereignisse wartet und diese in einen CloudWatch Log-Log-Stream schreibt.

Voraussetzungen

Dieses Tutorial unterstellt eine funktionierende Datenverarbeitungsumgebung samt Auftragswarteschlange, die bereit ist, Aufträge aufzunehmen. Wenn Sie keine laufende Rechenumgebung und keine Job-Warteschlange haben, aus der Sie Ereignisse erfassen können, folgen Sie den Schritten unter, um eine Erste Schritte mit AWS Batch Tutorials zu erstellen. Am Ende

dieses Tutorials können Sie optional einen Job an diese Job-Warteschlange senden, um zu testen, ob Sie Ihre Lambda-Funktion korrekt konfiguriert haben.

Themen

- Tutorial: Lambda-Funktion erstellen
- Tutorial: Registrieren Sie die Ereignisregel
- Tutorial: Testen Sie Ihre Konfiguration

Tutorial: Lambda-Funktion erstellen

In diesem Verfahren erstellen Sie eine einfache Lambda-Funktion, die als Ziel für AWS Batch Event-Stream-Nachrichten dient.

So erstellen Sie eine Lambda-Zielfunktion

1. Öffnen Sie die AWS Lambda Konsole unter. https://console.aws.amazon.com/lambda/
2. Wählen Sie Funktion erstellen und Von Grund auf neu erstellen aus.
3. Geben Sie für Function name (Funktionsname) batch-event-stream-handler ein.
4. Wählen Sie für Runtime (Laufzeit) die Option Python 3.8 aus.
5. Wählen Sie Funktion erstellen aus.
6. Bearbeiten Sie im Abschnitt Codequelle den Beispielcode so, dass er dem folgenden Beispiel entspricht:

```
import json


def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
 type of: aws.batch")

    print(json.dumps(event))
```

Dies ist eine einfache Python 3.8-Funktion, die die von gesendeten Ereignisse ausgibt AWS Batch. Wenn alles korrekt konfiguriert ist, werden die Ereignisdetails am Ende dieses Tutorials im CloudWatch Log-Protokollstream angezeigt, der dieser Lambda-Funktion zugeordnet ist.

7. Wählen Sie Bereitstellen.

Tutorial: Registrieren Sie die Ereignisregel

In diesem Abschnitt erstellen Sie eine EventBridge Ereignisregel, die Auftragsereignisse erfasst, die aus Ihren AWS Batch Ressourcen stammen. Diese Regel erfasst alle Ereignisse, die aus dem AWS Batch Konto stammen, in dem sie definiert ist. Die Jobnachrichten selbst enthalten Informationen über die Ereignisquelle, einschließlich der Job-Warteschlange, in der sie eingereicht wurden. Sie können diese Informationen verwenden, um Ereignisse programmgesteuert zu filtern und zu sortieren.

> **Note**
>
> Wenn Sie die verwenden, AWS Management Console um eine Ereignisregel zu erstellen, fügt die Konsole automatisch die IAM-Berechtigungen hinzu, EventBridge um Ihre Lambda-Funktion aufzurufen. Wenn Sie jedoch eine Ereignisregel mithilfe von erstellen AWS CLI, müssen Sie Berechtigungen explizit gewähren. Weitere Informationen finden Sie unter Ereignisse und Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

Um Ihre EventBridge Regel zu erstellen

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie als Event bus (Event Bus) den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel mit Ereignissen aus Ihrem eigenen Konto übereinstimmt, wählen Sie AWS -Standard-Event-Bus aus. Wenn ein AWS Service in Ihrem

Konto ein Ereignis ausgibt, wird dieses immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.

6. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
7. Wählen Sie Weiter.
8. Wählen Sie für Event source (Ereignisquelle) Other (Andere) aus.
9. Wählen Sie für Ereignismuster die Option Benutzerdefinierte Muster (JSON-Editor) aus.
10. Fügen Sie das folgende Ereignismuster in das Textfeld ein.

```
{
  "source": [
    "aws.batch"
  ]
}
```

Diese Regel gilt für alle Ihre AWS Batch Gruppen und für jedes AWS Batch Ereignis. Alternativ können Sie eine spezifischere Regel erstellen, damit bestimmte Ergebnisse gefiltert werden.

11. Wählen Sie Weiter.
12. Bei Zieltypen wählen Sie AWS -Service aus.
13. Wählen Sie für Select a target die Option Lambda-Funktion und wählen Sie Ihre Lambda-Funktion aus.
14. (Optional) Gehen Sie unter Additional settings (Weitere Einstellungen) wie folgt vor:

    a. Geben Sie für Maximum age of event (Maximales Alter des Ereignisses) einen Wert zwischen einer Minute (00:01) und 24 Stunden (24:00) ein.

    b. Geben Sie für Wiederholungsversuche eine Zahl zwischen 0 und 185 ein.

    c. Wählen Sie für Warteschlange für unzustellbare Briefe aus, ob Sie eine standardmäßige Amazon SQS SQS-Warteschlange als Warteschlange für unzustellbare Briefe verwenden möchten. EventBridge sendet Ereignisse, die dieser Regel entsprechen, an die Warteschlange für unzustellbare Briefe, wenn sie nicht erfolgreich an das Ziel zugestellt wurden. Führen Sie eine der folgenden Aktionen aus:

        - Klicken Sie auf Keine, um keine Warteschlange für unzustellbare Nachrichten zu verwenden.
        - Wählen Sie Wählen Sie eine Amazon SQS SQS-Warteschlange im aktuellen AWS Konto aus, die als Warteschlange für eingehende Briefe verwendet werden soll, und wählen Sie dann die zu verwendende Warteschlange aus der Drop-down-Liste aus.

- Wählen Sie Wählen Sie eine Amazon SQS SQS-Warteschlange in einem anderen AWS Konto als Warteschlange für unzustellbare Briefe aus und geben Sie dann den ARN der Warteschlange ein, die Sie verwenden möchten. Sie müssen der Warteschlange eine ressourcenbasierte Richtlinie hinzufügen, die das Senden von Nachrichten an die EventBridge Warteschlange ermöglicht. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erteilen von Berechtigungen für die Warteschlange mit unzustellbaren Briefen.

15. Wählen Sie Weiter.
16. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter EventBridge Amazon-Tags im EventBridge Amazon-Benutzerhandbuch.
17. Wählen Sie Weiter.
18. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Tutorial: Testen Sie Ihre Konfiguration

Sie können Ihre EventBridge Konfiguration jetzt testen, indem Sie einen Job an Ihre Job-Warteschlange senden. Wenn alles richtig konfiguriert ist, wird Ihre Lambda-Funktion ausgelöst und sie schreibt die Ereignisdaten in einen CloudWatch Log-Log-Stream für die Funktion.

So testen Sie die Konfiguration

1. Öffnen Sie die AWS Batch Konsole unter. https://console.aws.amazon.com/batch/
2. Reichen Sie einen neuen AWS Batch Job ein. Weitere Informationen finden Sie unter Tutorial: Einen Job einreichen.
3. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
4. Wählen Sie im Navigationsbereich Logs und wählen Sie die Protokollgruppe für Ihre Lambda-Funktion aus (z. B. ***my-function***/aws/lambda/).
5. Wählen Sie einen Protokollstream aus, um die Ereignisdaten anzuzeigen.

Tutorial: Senden von Amazon Simple Notification Service-Benachrichtigungen für fehlgeschlagene Auftragsereignisse

In diesem Tutorial konfigurieren Sie eine EventBridge Amazon-Ereignisregel, die nur Auftragsereignisse erfasst, bei denen der Job in einen `FAILED` Status übergegangen ist. Am Ende

dieses Tutorials können Sie optional auch einen Job an diese Job-Warteschlange senden. Dies dient dazu, zu testen, ob Sie Ihre Amazon SNS SNS-Benachrichtigungen korrekt konfiguriert haben.

Voraussetzungen

Dieses Tutorial unterstellt eine funktionierende Datenverarbeitungsumgebung samt Auftragswarteschlange, die bereit ist, Aufträge aufzunehmen. Wenn Sie keine laufende Rechenumgebung und keine Job-Warteschlange haben, aus der Sie Ereignisse erfassen können, folgen Sie den Schritten unter, um eine Erste Schritte mit AWS Batch Tutorials zu erstellen.

Themen

- Tutorial: Ein Amazon SNS SNS-Thema erstellen und abonnieren
- Tutorial: Registrieren Sie eine Ereignisregel
- Tutorial: Testen Sie Ihre Regel
- Alternative Regel: Batch-Job-Warteschlange blockiert

Tutorial: Ein Amazon SNS SNS-Thema erstellen und abonnieren

Mit diesem Tutorial konfigurieren Sie ein Amazon SNS-Thema, das als Ereignisziel für Ihre neue Ereignisregel dient.

Erstellen eines Amazon SNS-Themas

1. Öffnen Sie die Amazon SNS SNS-Konsole unter https://console.aws.amazon.com/sns/v3/home.
2. Wählen Sie Themen, Thema erstellen aus.
3. Wählen Sie unter Type (Typ) die Option Standard aus.
4. Geben Sie als Namen Thema ein **`JobFailedAlert`** und wählen Sie Create topic aus.
5. Wählen Sie auf dem JobFailedAlertBildschirm die Option Abonnement erstellen aus.
6. Wählen Sie unter Protocol (Protokoll) die Option Email (E-Mail) aus.
7. Geben Sie für Endpunkt eine E-Mail-Adresse ein, auf die Sie aktuell Zugriff haben, und wählen Sie Abonnement erstellen aus.
8. Überprüfen Sie Ihr E-Mail-Konto und warten Sie auf eine E-Mail-Nachricht zur Bestätigung Ihres Abonnements. Wenn Sie sie erhalten, wählen Sie Confirm Abonnement aus.

Tutorial: Registrieren Sie eine Ereignisregel

Registrieren Sie nun eine Ereignisregel, mit der nur Ereignisse zu fehlgeschlagenen Aufträgen erfasst werden.

Um deine EventBridge Regel zu registrieren

1. Öffnen Sie die EventBridge Amazon-Konsole unter https://console.aws.amazon.com/events/.
2. Wählen Sie im Navigationsbereich Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für die Regel ein.

   Eine Regel darf nicht denselben Namen wie eine andere Regel in derselben Region und auf demselben Event Bus haben.
5. Wählen Sie als Event bus (Event Bus) den Event Bus aus, den Sie dieser Regel zuordnen möchten. Wenn Sie möchten, dass diese Regel mit Ereignissen aus Ihrem eigenen Konto übereinstimmt, wählen Sie AWS -Standard-Event-Bus aus. Wenn ein AWS Service in Ihrem Konto ein Ereignis ausgibt, wird dieses immer an den Standard-Event-Bus Ihres Kontos weitergeleitet.
6. Bei Regeltyp wählen Sie Regel mit einem Ereignismuster aus.
7. Wählen Sie Weiter.
8. Wählen Sie für Event source (Ereignisquelle) Other (Andere) aus.
9. Wählen Sie für Ereignismuster die Option Benutzerdefinierte Muster (JSON-Editor) aus.
10. Fügen Sie das folgende Ereignismuster in das Textfeld ein.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
```

```
}
```

Dieser Code definiert eine EventBridge Regel, die für jedes Ereignis gilt, bei dem sich der Jobstatus befindet`FAILED`. Weitere Informationen zu Ereignismustern finden Sie unter Ereignisse und Ereignismuster im EventBridge Amazon-Benutzerhandbuch.

11. Wählen Sie Weiter.
12. Bei Zieltypen wählen Sie AWS -Service aus.
13. Wählen Sie für Wählen Sie ein Ziel die Option SNS-Thema und für Thema die Option JobFailedAlert.
14. (Optional) Gehen Sie unter Additional settings (Weitere Einstellungen) wie folgt vor:
    a. Geben Sie für Maximum age of event (Maximales Alter des Ereignisses) einen Wert zwischen einer Minute (00:01) und 24 Stunden (24:00) ein.
    b. Geben Sie für Wiederholungsversuche eine Zahl zwischen 0 und 185 ein.
    c. Wählen Sie für Warteschlange für unzustellbare Briefe aus, ob Sie eine standardmäßige Amazon SQS SQS-Warteschlange als Warteschlange für unzustellbare Briefe verwenden möchten. EventBridge sendet Ereignisse, die dieser Regel entsprechen, an die Warteschlange für unzustellbare Briefe, wenn sie nicht erfolgreich an das Ziel zugestellt wurden. Führen Sie eine der folgenden Aktionen aus:
        - Klicken Sie auf Keine, um keine Warteschlange für unzustellbare Nachrichten zu verwenden.
        - Wählen Sie Wählen Sie eine Amazon SQS SQS-Warteschlange im aktuellen AWS Konto aus, die als Warteschlange für eingehende Briefe verwendet werden soll, und wählen Sie dann die zu verwendende Warteschlange aus der Drop-down-Liste aus.
        - Wählen Sie Wählen Sie eine Amazon SQS SQS-Warteschlange in einem anderen AWS Konto als Warteschlange für unzustellbare Briefe aus und geben Sie dann den ARN der Warteschlange ein, die Sie verwenden möchten. Sie müssen der Warteschlange eine ressourcenbasierte Richtlinie beifügen, die das Senden von Nachrichten an die EventBridge Warteschlange ermöglicht. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch unter Erteilen von Berechtigungen für die Warteschlange mit unzustellbaren Briefen.
15. Wählen Sie Weiter.
16. (Optional) Geben Sie ein oder mehrere Tags für die Regel ein. Weitere Informationen finden Sie unter EventBridge Amazon-Tags im EventBridge Amazon-Benutzerhandbuch.

17. Wählen Sie Weiter.
18. Überprüfen Sie die Details der Regel und wählen Sie dann Regel erstellen aus.

Tutorial: Testen Sie Ihre Regel

Übermitteln Sie zum Testen der Regel einen Auftrag, der kurz nach dem Start mit einem Beendigungscode ungleich Null beendet wird. Wenn Ihre Ereignisregel korrekt konfiguriert ist, sollten Sie innerhalb weniger Minuten eine E-Mail-Nachricht mit dem Ereignistext erhalten.

So testen Sie eine Regel

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Reichen Sie einen neuen AWS Batch Job ein. Weitere Informationen finden Sie unter Tutorial: Einen Job einreichen. Ersetzen Sie den Befehl des Auftrags durch diesen Befehl, damit der Container mit einem Beendigungscode von 1 beendet wird.

   ```
   /bin/sh, -c, 'exit 1'
   ```

3. Überprüfen Sie Ihre E-Mails, um zu bestätigen, dass Sie eine E-Mail-Benachrichtigung für die Benachrichtigung über den fehlgeschlagenen Job erhalten haben.

Alternative Regel: Batch-Job-Warteschlange blockiert

Um eine Ereignisregel zu erstellen, die überwacht, ob die Batch-Job-Warteschlange blockiert ist, wiederholen Sie diese Tutorials mit den folgenden Änderungen:

1. In Tutorial: Ein Amazon SNS SNS-Thema erstellen und abonnieren, `BlockedJobQueue` als Themennamen verwenden.
2. Verwenden Sie in Tutorial: Registrieren Sie eine Ereignisregel das folgende Muster im JSON-Editor:

   ```
   {
      "detail-type": [
        "Batch Job Queue Blocked"
      ],
      "source": [
        "aws.batch"
      ]
   ```

```
}
```

Elastic Fabric Adapter

Ein Elastic Fabric Adapter (EFA) ist ein Netzwerkgerät zur Beschleunigung von High Performance Computing(HPC)-Anwendungen. AWS Batch unterstützt Anwendungen, die EFA verwenden, wenn die folgenden Bedingungen erfüllt sind.

- Eine Liste der unterstützten Instance-Typen finden Sie unter Unterstützte Instance-Typen im EC2 Amazon-Benutzerhandbuch. EFAs

> **Tip**
>
> Um eine Liste der Instance-Typen anzuzeigen, die EFAs Inan unterstützen AWS-Region, führen Sie den folgenden Befehl aus. Vergleichen Sie dann die zurückgegebene Liste mit der Liste der verfügbaren Instanztypen in der AWS Batch Konsole.
>
> ```
> $ aws ec2 describe-instance-types  --region us-east-1  --filters Name=network-info.efa-supported,Values=true  --query "InstanceTypes[*].[InstanceType]"  --output text | sort
> ```

- Eine Liste der Betriebssysteme, die EFA unterstützen, finden Sie unter Unterstützte Betriebssysteme.
- Der EFA-Treiber im AMI ist geladen.
- Die Sicherheitsgruppe für den EFA muss allen eingehenden und ausgehenden Datenverkehr von und zu der Sicherheitsgruppe selbst zulassen.
- Alle Instances, die eine EFA verwenden, müssen sich in derselben Cluster-Platzierungsgruppe befinden.
- Die Auftragsdefinition muss ein `devices`-Mitglied enthalten, dessen `hostPath` auf `/dev/infiniband/uverbs0` festgelegt ist, damit das EFA-Gerät an den Container übergeben wird. Wenn `containerPath` angegeben, muss es auch auf gesetzt werden. `/dev/infiniband/uverbs0` Wenn `permissions` festgelegt ist, muss es auf `READ` | `WRITE` | `MKNOD` festgelegt werden.

  Die Standorte der LinuxParametersMitglieder unterscheiden sich bei parallel Jobs mit mehreren Knoten und Container-Jobs mit einem Knoten. Die folgenden Beispiele zeigen die Unterschiede, es fehlen jedoch erforderliche Werte.

Example Beispiel für einen parallelen Auftrag mit mehreren Knoten

```
{
  "jobDefinitionName": "EFA-MNP-JobDef",
  "type": "multinode",
  "nodeProperties": {
    ...
    "nodeRangeProperties": [
      {
        ...
        "container": {
          ...
          "linuxParameters": {
            "devices": [
              {
                "hostPath": "/dev/infiniband/uverbs0",
                "containerPath": "/dev/infiniband/uverbs0",
                "permissions": [
                    "READ", "WRITE", "MKNOD"
                ]
              },
            ],
          },
        },
      },
    ],
  },
}
```

Example Beispiel für einen Container-Auftrag mit Einzelknoten

```
{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
```

```
        ],
      },
    },
  }
```

Weitere Informationen zu EFA finden Sie unter Elastic Fabric Adapter im EC2 Amazon-Benutzerhandbuch.

Überwachen AWS Batch

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit AWS Batch und Leistung Ihrer AWS Lösung.

Wir empfehlen Ihnen dringend, Überwachungsdaten aus allen Teilen Ihrer AWS Lösung zu sammeln, um das Debuggen eines etwaigen Fehlers an mehreren Stellen zu erleichtern. Erstellen Sie zunächst einen Überwachungsplan, der die folgenden Fragen beantwortet. Wenn Sie sich nicht sicher sind, wie Sie diese beantworten sollen, können Sie trotzdem Amazon CloudWatch Logs verwenden, um Ihre Leistungsbasislinien festzulegen.

- Was sind Ihre Überwachungsziele?
- Welche Ressourcen möchten Sie überwachen?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungs-Tools möchten Sie verwenden?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Ihr nächster Schritt besteht darin, einen Basiswert für die normale AWS Batch Leistung in Ihrer Umgebung festzulegen, indem Sie die Leistung zu verschiedenen Zeiten und unter verschiedenen Lastbedingungen messen. Bewahren Sie bei der Überwachung historische Überwachungsdaten auf, damit Sie sie mit aktuellen Leistungsdaten vergleichen können. AWS Batch Auf diese Weise können Sie normale Leistungsmuster und Leistungsanomalien identifizieren und Methoden zu deren Handhabung entwickeln.

Die Themen in diesem Abschnitt dienen als Einstieg in die Protokollierung und Überwachung von AWS Batch.

Themen

- Verwenden von CloudWatch Protokollen mit AWS Batch
- AWS Batch CloudWatch Einblicke in Container
- Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen

Verwenden von CloudWatch Protokollen mit AWS Batch

Sie können Ihre AWS Batch Jobs auf EC2 Ressourcen so konfigurieren, dass detaillierte Protokollinformationen und Metriken an CloudWatch Logs gesendet werden. Auf diese Weise können Sie verschiedene Protokolle Ihrer Jobs an einem zentralen Ort einsehen. Weitere Informationen zu CloudWatch Logs finden Sie unter Was ist Amazon CloudWatch Logs? im CloudWatch Amazon-Benutzerhandbuch.

> **Note**
>
> Standardmäßig ist CloudWatch Logs für AWS Fargate-Container aktiviert.

Um die CloudWatch Log-Protokollierung zu aktivieren und anzupassen, überprüfen Sie die folgenden einmaligen Konfigurationsaufgaben:

- Für AWS Batch Rechenumgebungen, die auf EC2 Ressourcen basieren, fügen Sie der `ecsInstanceRole` Rolle eine IAM-Richtlinie hinzu. Weitere Informationen finden Sie unter the section called “Tutorial: Fügen Sie eine CloudWatch Logs-IAM-Richtlinie hinzu”.
- Erstellen Sie eine EC2 Amazon-Startvorlage, die eine detaillierte CloudWatch Überwachung beinhaltet, und geben Sie die Vorlage dann an, wenn Sie Ihre AWS Batch Rechenumgebung erstellen. Sie können den CloudWatch Agenten auch auf einem vorhandenen Image installieren und das Image dann im Assistenten für die AWS Batch erste Ausführung angeben.
- (Optional) Konfigurieren Sie den awslogs-Treiber. Sie können Parameter hinzufügen, die das Standardverhalten sowohl EC2 für Fargate-Ressourcen als auch für Fargate-Ressourcen ändern. Weitere Informationen finden Sie unter the section called “Verwenden Sie den awslogs-Protokolltreiber”.

Themen

- Tutorial: Fügen Sie eine CloudWatch Logs-IAM-Richtlinie hinzu
- Installieren und konfigurieren Sie den CloudWatch Agenten
- Tutorial: CloudWatch Protokolle anzeigen

Tutorial: Fügen Sie eine CloudWatch Logs-IAM-Richtlinie hinzu

Bevor Ihre Jobs Protokolldaten und detaillierte Metriken an CloudWatch Logs senden können, müssen Sie eine IAM-Richtlinie erstellen, die die Logs verwendet. CloudWatch APIs Nachdem Sie die IAM-Richtlinie erstellt haben, fügen Sie sie der `ecsInstanceRole` Rolle hinzu.

> **Note**
>
> Wenn die `ECS-CloudWatchLogs` Richtlinie nicht an die `ecsInstanceRole` Rolle angehängt ist, können grundlegende Metriken trotzdem an CloudWatch Logs gesendet werden. Die Basiskennzahlen enthalten jedoch keine Protokolldaten oder detaillierte Messwerte wie den freien Festplattenspeicher.

AWS Batch Rechenumgebungen verwenden EC2 Amazon-Ressourcen. Wenn Sie mit dem Assistenten für die AWS Batch erste Ausführung eine Rechenumgebung erstellen, AWS Batch erstellt er die `ecsInstanceRole` Rolle und konfiguriert die Umgebung damit.

Wenn Sie den Assistenten für die erste Ausführung nicht verwenden, können Sie die `ecsInstanceRole` Rolle angeben, wenn Sie eine Rechenumgebung in der AWS Command Line Interface OR-API erstellen. AWS Batch Weitere Informationen finden Sie in der AWS CLI Befehlsreferenz oder AWS Batch API-Referenz.

So erstellen Sie die **`ECS-CloudWatchLogs`**-IAM-Richtlinie

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie JSON und geben Sie dann die folgende Richtlinie ein:

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        }
    ]
}
```

5. Wählen Sie Weiter: Tags aus.
6. (Optional) Wählen Sie unter Tags hinzufügen die Option Tag hinzufügen aus, um der Richtlinie ein Tag hinzuzufügen.
7. Wählen Sie Weiter: Prüfen aus.
8. Geben Sie auf der Seite „Richtlinie überprüfen“ als Namen eine Beschreibung ein**`ECS-CloudWatchLogs`**, und geben Sie dann optional eine Beschreibung ein.
9. Wählen Sie Richtlinie erstellen aus.

So fügen Sie die Richtlinie **`ECS-CloudWatchLogs`** an die **`ecsInstanceRole`** an

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen.
3. Wählen Sie `ecsInstanceRole`. Wenn die Rolle nicht existiert, folgen Sie den Verfahren unter Amazon ECS-Instance-Rolle So erstellen Sie die Rolle.
4. Wählen Sie „Berechtigungen hinzufügen“ und anschließend „Richtlinien anhängen“.
5. Wählen Sie die CloudWatchLogsECS-Richtlinie und dann die Option Richtlinie anhängen aus.

Installieren und konfigurieren Sie den CloudWatch Agenten

Sie können eine EC2 Amazon-Startvorlage erstellen, die die CloudWatch Überwachung beinhaltet. Weitere Informationen finden Sie unter Starten einer Instance über eine Startvorlage und Erweiterte Details im EC2 Amazon-Benutzerhandbuch.

Sie können den CloudWatch Agenten auch auf einem vorhandenen Amazon EC2 AMI installieren und dann das Image im Assistenten für die AWS Batch erste Ausführung angeben. Weitere

Informationen finden Sie unter Installation des CloudWatch Agenten undErste Schritte mit AWS Batch Tutorials.

> **Note**
>
> Startvorlagen werden auf AWS Fargate Ressourcen nicht unterstützt.

Tutorial: CloudWatch Protokolle anzeigen

Sie können CloudWatch Log-Logs in der anzeigen und durchsuchen AWS Management Console.

> **Note**
>
> Es kann einige Minuten dauern, bis Daten in CloudWatch Logs angezeigt werden.

Um Ihre CloudWatch Logs-Daten einzusehen

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich Logs und anschließend Log-Gruppen aus.

3. Wählen Sie eine Protokollgruppe aus, die angezeigt werden soll.

4. Wählen Sie einen Protokollstream aus, der angezeigt werden soll. Standardmäßig werden die Streams durch die ersten 200 Zeichen des Jobnamens und der Amazon ECS-Aufgaben-ID identifiziert.

> **Tip**
>
> Um Log-Stream-Daten herunterzuladen, wählen Sie Aktionen.

AWS Batch CloudWatch Einblicke in Container

CloudWatch Container Insights sammelt, aggregiert und fasst Metriken und Protokolle aus Ihren AWS Batch Rechenumgebungen und Jobs zusammen. Die Metriken umfassen CPU-, Arbeitsspeicher-, Festplatten- und Netzwerkauslastung. Sie können diese Metriken zu CloudWatch Dashboards hinzufügen.

Betriebliche Daten werden als Performance-Protokollereignisse gesammelt. Dabei handelt es sich um Einträge mit einem strukturierten JSON-Schema, das die Aufnahme und Speicherung von Daten hoher Kardinalität in in großem Umfang ermöglicht. Aus diesen Daten werden aggregierte Metriken auf höherer Ebene auf der Rechenumgebung und auf Jobebene als Metriken CloudWatch erstellt. CloudWatch Weitere Informationen finden Sie unter Container Insights Structured Logs for Amazon ECS im CloudWatch Amazon-Benutzerhandbuch.

> **Important**
>
> CloudWatch Container Insights werden als benutzerdefinierte Metriken von berechnet CloudWatch. Weitere Informationen finden Sie unter Amazon CloudWatch Events-Preise

Themen

- Tutorial: Container Insights einschalten

Tutorial: Container Insights einschalten

Gehen Sie wie folgt vor, um Container Insights für AWS Batch Rechenumgebungen zu aktivieren.

1. Öffnen Sie die AWS Batch -Konsole.
2. Wählen Sie Compute Environments aus.
3. Wählen Sie die gewünschte Rechenumgebung aus.
4. Aktivieren Sie für Container Insights Container Insights für die Rechenumgebung.

> **Tip**
>
> Sie können ein Standardintervall auswählen, um die Metriken zu aggregieren, oder ein benutzerdefiniertes Intervall erstellen.

Standardmäßig werden die folgenden Metriken angezeigt. Eine vollständige Liste der Amazon ECS Container Insights-Metriken finden Sie unter Amazon ECS Container Insights-Metriken im CloudWatch Amazon-Benutzerhandbuch.

- **`JobCount`**— Die Anzahl der Jobs, die in der Rechenumgebung ausgeführt werden.
- **`ContainerInstanceCount`**— Die Anzahl der Amazon Elastic Compute Cloud-Instances, auf denen der Amazon ECS-Agent ausgeführt wird und die in der Rechenumgebung registriert sind.
- **`MemoryReserved`**— Der Speicher, der für Jobs in der Rechenumgebung reserviert ist. Diese Metrik wird nur für Jobs erfasst, für die in ihrer Auftragsdefinition eine definierte Speicherreservierung enthalten ist.
- **`MemoryUtilized`**— Der Speicher, der von Jobs in der Rechenumgebung verwendet wird. Diese Metrik wird nur für Jobs erfasst, deren Jobdefinition eine definierte Speicherreservierung enthält.
- **`CpuReserved`**— Die CPU-Einheiten, die für Jobs in der Rechenumgebung reserviert sind. Diese Metrik wird nur für Jobs erfasst, deren Jobdefinition eine definierte CPU-Reservierung enthält.
- **`CpuUtilized`**— Die CPU-Einheiten, die von Jobs in der Rechenumgebung verwendet werden. Diese Metrik wird nur für Jobs erfasst, deren Jobdefinition eine definierte CPU-Reservierung enthält.
- **`NetworkRxBytes`**- Die Anzahl der empfangenen Byte. Diese Metrik ist nur für Container in Jobs verfügbar, die den Netzwerkmodus `awsvpc` oder den Bridge-Netzwerkmodus verwenden.
- **`NetworkTxBytes`**— Die Anzahl der übertragenen Byte. Diese Metrik ist nur für Container in Jobs verfügbar, die den Netzwerkmodus `awsvpc` oder den Bridge-Netzwerkmodus verwenden.
- **`StorageReadBytes`**— Die Anzahl der Byte, die aus dem Speicher gelesen werden.
- **`StorageWriteBytes`**— Die Anzahl der Byte, die in den Speicher geschrieben werden.

Verwenden Sie CloudWatch Logs, um AWS Batch Amazon EKS-Jobs zu überwachen

Sie können Amazon CloudWatch Logs verwenden, um all Ihre Protokolldateien an einem Ort zu überwachen, zu speichern und anzuzeigen. Mithilfe von CloudWatch Logs können Sie Protokolldaten aus mehreren Quellen suchen, filtern und analysieren.

Sie können ein Formular AWS herunterladen Fluent Bit Bild, das ein Plugin zur Überwachung von AWS Batch Amazon EKS-Jobs in CloudWatch Logs enthält. Fluent Bit ist ein Open-Source-Protokollprozessor und Forwarder, der sowohl Docker als auch Kubernetes kompatibel. Wir

empfehlen die Verwendung Fluent Bit als Ihren Log-Router, weil er weniger ressourcenintensiv ist als Fluentd. Weitere Informationen finden Sie unter Installieren des CloudWatch Agenten mit dem Amazon CloudWatch Observability EKS-Add-on oder dem Helm-Diagramm.

Voraussetzungen

- Hängen Sie die `CloudWatchAgentServerPolicy` Richtlinie an die AWS Identity and Access Management Richtlinie Ihrer Worker-Knoten an. Weitere Informationen finden Sie unter Voraussetzungen überprüfen.

Installieren Sie das Add-on

Für Anweisungen zur Installation AWS für Fluent Bit und erstellen Sie die CloudWatch Gruppen, siehe Installieren Sie den CloudWatch Agenten mit dem Amazon CloudWatch Observability EKS-Add-on oder dem Helm-Diagramm.

Bei der Installation des Add-ons müssen Sie die folgenden zusätzlichen Konfigurationsdaten angeben:

- Wenn Sie das Add-on zusammen mit dem installieren, müssen AWS Management Console Sie die folgenden Toleranzen bei den Konfigurationswerten angeben:

```
{
  "tolerations": [
    {
      "key": "batch.amazonaws.com/batch-node",
      "operator": "Exists"
    }
  ]
}
```

- Wenn Sie das Add-on mit dem installieren, AWS CLI fügen Sie die folgenden Argumente hinzu:

```
--configuration-values '{"tolerations":[{"key":"batch.amazonaws.com/batch-
node","operator":"Exists"}]}'
```

> **Tip**
>
> Denken Sie daran Fluent Bit verwendet 2,5 CPUs und 100 MB Speicher auf AWS Batch Knoten. Dadurch wird die insgesamt verfügbare Kapazität für AWS Batch Jobs reduziert. Berücksichtigen Sie dies, wenn Sie die Größe Ihrer Jobs festlegen.

Kennzeichnen Sie Ihre AWS Batch Ressourcen

Um Ihnen bei der Verwaltung Ihrer AWS Batch Ressourcen zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. In diesem Thema werden Tags (Markierungen) und deren Erstellung beschrieben.

Themen

- Grundlagen zu Tags (Markierungen)
- Markieren Ihrer -Ressourcen mit Tags (Markierungen)
- Tag-Einschränkungen
- Tutorial: Tags mit der Konsole verwalten
- Tags mit der CLI oder API verwalten

Grundlagen zu Tags (Markierungen)

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource zuweisen. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen.

Mithilfe von Tags können Sie Ihre AWS Ressourcen beispielsweise nach Zweck, Eigentümer oder Umgebung kategorisieren. Wenn Sie viele Ressourcen desselben Typs haben, können Sie bestimmte Ressourcen basierend auf den zugewiesenen Tags schnell bestimmen. Sie können beispielsweise eine Reihe von Tags für Ihre AWS Batch Services definieren, um Ihnen zu helfen, den Besitzer und die Stack-Ebene der einzelnen Services nachzuverfolgen. Sie sollten für jeden Ressourcentyp einen konsistenten Satz von Tag-Schlüsseln entwickeln.

Tags werden nicht automatisch Ihren Ressourcen zugewiesen. Nachdem Sie ein Tag hinzugefügt haben, können Sie jederzeit Tag-Schlüssel und -Werte bearbeiten oder Tags aus einer Ressource entfernen. Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.

Tags haben keine semantische Bedeutung AWS Batch und werden ausschließlich als Zeichenfolge interpretiert. Sie können den Wert eines Tags (Markierung) zwar auf eine leere Zeichenfolge, jedoch nicht null festlegen. Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.

Sie können mit Tags arbeiten, indem Sie die AWS Management Console AWS CLI, und die AWS Batch API verwenden.

Wenn Sie AWS Identity and Access Management (IAM) verwenden, können Sie steuern, welche Benutzer in Ihrem AWS Konto berechtigt sind, Tags zu erstellen, zu bearbeiten oder zu löschen.

Markieren Ihrer -Ressourcen mit Tags (Markierungen)

Sie können neue oder bestehende AWS Batch Computerumgebungen, Jobs, Jobdefinitionen, Jobwarteschlangen und Planungsrichtlinien taggen.

Wenn Sie die AWS Batch Konsole verwenden, können Sie Tags auf neue Ressourcen anwenden, wenn diese erstellt werden, oder jederzeit auf bestehende Ressourcen, indem Sie die Registerkarte „Tags“ auf der entsprechenden Ressourcenseite verwenden.

Wenn Sie die AWS Batch API, das AWS CLI oder ein AWS SDK verwenden, können Sie mithilfe des `tags` Parameters der entsprechenden API-Aktion Tags auf neue Ressourcen oder mithilfe der `TagResource` API-Aktion auf vorhandene Ressourcen anwenden. Weitere Informationen finden Sie unter TagResource.

Bei einigen Aktionen zur Ressourcenerstellung können Sie Tags für eine Ressource angeben, wenn die Ressource erstellt wird. Wenn Tags während der Ressourcenerstellung nicht angewendet werden können, schlägt die Ressourcenerstellung fehl. Auf diese Weise wird sichergestellt, dass Ressourcen, die Sie bei der Erstellung markieren möchten, entweder mit angegebenen Tags oder gar nicht erstellt werden. Wenn Sie Ressourcen zum Zeitpunkt der Erstellung markieren, müssen Sie nach der Ressourcenerstellung keine benutzerdefinierten Tagging-Skripts ausführen.

In der folgenden Tabelle werden die AWS Batch Ressourcen beschrieben, die markiert werden können, und die Ressourcen, die bei der Erstellung markiert werden können.

Tag-Unterstützung für AWS Batch Ressourcen

Ressource	Unterstützt Tags (Markierungen)	Unterstützt Tag-Propagierung	Unterstützt das Taggen bei der Erstellung (AWS Batch API AWS CLI, AWS SDK)
AWS Batch Computerumgebungen	Ja	Nein. Compute-Umgebungs-Tags werden nicht auf andere Ressourcen übertragen. Die Tags	Ja

Ressource	Unterstützt Tags (Markierungen)	Unterstützt Tag-Propagierung	Unterstützt das Taggen bei der Erstellung (AWS Batch API AWS CLI, AWS SDK)
		für die Ressourcen werden im Tags-Element des ComputeResources-Objekts angegeben, das im API-Vorgang übergeben wurde. CreateComputeEnvironment	
AWS Batch Jobs	Ja	Ja	Ja
AWS Batch Berufsdefinitionen	Ja	Nein	Ja
AWS Batch Job-Warteschlangen	Ja	Nein	Ja
AWS Batch Richtlinien für die Terminplanung	Ja	Nein	Ja

Tag-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Maximale Anzahl von Tags (Markierungen) pro Ressource: 50
- Jeder Tag (Markierung) muss für jede Ressource eindeutig sein. Jeder Tag (Markierung) kann nur einen Wert haben.
- Maximale Schlüssellänge: 128 Unicode-Zeichen in UTF-8
- Maximale Wertlänge: 256 Unicode-Zeichen in UTF-8

- Wenn Ihr Tagging-Schema für mehrere AWS Dienste und Ressourcen verwendet wird, denken Sie daran, dass für andere Dienste möglicherweise Einschränkungen hinsichtlich der zulässigen Zeichen gelten. Allgemein erlaubte Zeichen sind Buchstaben, Zahlen, Leerzeichen, die in UTF-8 darstellbar sind, sowie die folgenden Zeichen: + - = . _ : / @.
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden.
- Verwenden `aws:` Sie weder für Schlüssel noch für Werte eine Kombination aus Groß- oder Kleinbuchstaben, z. B. ein Präfix, da es für AWS die Verwendung reserviert ist. `AWS:` Sie können keine Tag-Schlüssel oder -Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht auf Ihr tags-per-resource Limit angerechnet.

Tutorial: Tags mit der Konsole verwalten

Mithilfe der AWS Batch Konsole können Sie die Tags verwalten, die neuen oder vorhandenen Computerumgebungen, Jobs, Jobdefinitionen und Jobwarteschlangen zugeordnet sind.

Fügen Sie bei der Erstellung Tags zu einer einzelnen Ressource hinzu

Sie können Tags zu AWS Batch Computerumgebungen, Jobs, Jobdefinitionen, Jobwarteschlangen und Planungsrichtlinien hinzufügen, wenn Sie sie erstellen.

Hinzufügen und Löschen von Tags (Markierungen) für einzelne Ressourcen

AWS Batch ermöglicht es Ihnen, mit Ihren Clustern verknüpfte Tags direkt von der Seite der Ressource aus hinzuzufügen oder zu löschen.

So fügen Sie ein Tag zu einer einzelnen Ressource hinzu oder löschen es

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die Region aus, die Sie verwenden möchten.
3. Wählen Sie im Navigationsbereich einen Ressourcentyp aus (z. B. Job Queues).
4. Wählen Sie eine bestimmte Ressource aus und klicken Sie dann auf Tags bearbeiten.
5. Füge deine Tags nach Bedarf hinzu oder lösche sie.
   - Um ein Tag hinzuzufügen, geben Sie den Schlüssel und den Wert in den leeren Textfeldern am Ende der Liste an.
   - Um ein Tag zu löschen, klicken Sie

Delete icon
auf die Schaltfläche neben dem Tag.

6. Wiederholen Sie diesen Vorgang für jedes Tag, das Sie hinzufügen oder löschen möchten, und wählen Sie dann Tags bearbeiten, um den Vorgang abzuschließen.

Tags mit der CLI oder API verwalten

Verwenden Sie die folgenden AWS CLI Befehle oder AWS Batch API-Operationen, um die Tags für Ihre Ressourcen hinzuzufügen, zu aktualisieren, aufzulisten und zu löschen.

Tag-Unterstützung für AWS Batch Ressourcen

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Fügen Sie einen oder mehrere Tags hinzu oder überschreiben Sie sie.	TagResource	tag-resource	Hinzufügen — BATResourceTag
Löschen Sie ein oder mehrere Tags.	UntagResource	untag-resource	Entfernen — BATResourceTag
Listet Tags für eine Ressource auf	ListTagsForResource	list-tags-for-resource	Abrufen — BATResourceTag

Die folgenden Beispiele zeigen, wie man Tags an Ressourcen mithilfe der AWS CLI hinzufügt oder entfernt.

Beispiel 1: Markieren einer vorhandenen Ressource

Der folgende Befehl markiert eine vorhandene Ressource.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Beispiel 2: Entfernen der Markierung einer vorhandenen Ressource

Der folgende Befehl löscht ein Tag aus einer vorhandenen Ressource.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Beispiel 3: Tags für eine Ressource auflisten

Der folgende Befehl listet die Tags auf, die einer vorhandenen Ressource zugeordnet sind.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Mit einigen Aktionen zur Ressourcenerstellung können Sie Tags beim Erstellen der Ressource angeben. Die folgenden Aktionen unterstützen das Markieren bei der Erstellung.

Aufgabe	API-Aktion	AWS CLI	AWS Tools for Windows PowerShell
Erstellen Sie eine Rechenumgebung	CreateComputeEnvironment	create-compute-environment	Neu — BATCompute Umgebung
Erstellen Sie eine Job-Warteschlange	CreateJobQueue	create-job-queue	Neu — BATJob Warteschlange
Erstellen Sie eine Planungsrichtlinie	CreateSchedulingPolicy	create-scheduling-policy	Neu — BATScheduling Richtlinie
Registrieren Sie eine Jobdefinition	RegisterJobDefinition	register-job-definition	Registrieren — BATJob Definition
Übermitteln eines Auftrags	SubmitJob	Job einreichen	Einreichen- BATJob
Erstellen Sie eine verbrauchbare Ressource	CreateConsumableResource	create-consumable-resource	Erstellen — Ressource BATConsumable

Bewährte Methoden für AWS Batch

Sie können AWS Batch damit eine Vielzahl anspruchsvoller Rechen-Workloads in großem Umfang ausführen, ohne eine komplexe Architektur verwalten zu müssen. AWS Batch Jobs können in einer Vielzahl von Anwendungsfällen in Bereichen wie Epidemiologie, Gaming und maschinelles Lernen eingesetzt werden.

In diesem Thema werden die bewährten Methoden behandelt, die Sie bei der Verwendung beachten sollten, AWS Batch sowie Anleitungen zur Ausführung und Optimierung Ihrer Workloads bei der Verwendung. AWS Batch

Themen

- Wann sollten Sie es verwenden AWS Batch
- Checkliste für die skalierbare Ausführung
- Optimieren Sie Container und AMIs
- Wählen Sie die richtige Ressource für die Rechenumgebung
- Amazon EC2 On-Demand oder Amazon EC2 Spot
- Verwenden Sie die Best Practices von Amazon EC2 Spot für AWS Batch
- Häufige Fehler und Problembehebung

Wann sollten Sie es verwenden AWS Batch

AWS Batch führt Jobs in großem Umfang und zu niedrigen Kosten aus und bietet Warteschlangendienste und kostenoptimierte Skalierung. Allerdings ist nicht jeder Workload für die Ausführung mit geeignet. AWS Batch

- Kurze Jobs — Wenn ein Job nur wenige Sekunden lang ausgeführt wird, kann der Aufwand für die Planung des Batch-Jobs länger dauern als die Laufzeit des Jobs selbst. Um das Problem zu umgehen binpack Ihre Aufgaben zusammen, bevor Sie sie einreichen. AWS Batch Konfigurieren Sie dann Ihre AWS Batch Jobs so, dass sie die Aufgaben iterieren. Stellen Sie beispielsweise die einzelnen Aufgabenargumente in einer Amazon DynamoDB-Tabelle oder als Datei in einem Amazon S3 S3-Bucket bereit. Erwägen Sie, Aufgaben so zu gruppieren, dass die Jobs jeweils 3—5 Minuten laufen. Nach dir binpack Die Jobs, durchgehen Sie Ihre Aufgabengruppen innerhalb Ihres AWS Batch Jobs.

- Jobs, die sofort ausgeführt werden müssen — Jobs AWS Batch können schnell verarbeitet werden. Es AWS Batch ist jedoch ein Planer und optimiert das Preis-Leistungs-Verhältnis, die Priorität von Aufträgen und den Durchsatz. AWS Batch kann einige Zeit in Anspruch nehmen, um Ihre Anfragen zu bearbeiten. Wenn Sie innerhalb weniger Sekunden eine Antwort benötigen, ist ein servicebasierter Ansatz mit Amazon ECS oder Amazon EKS besser geeignet.

Checkliste für die skalierbare Ausführung

Bevor Sie eine große Arbeitslast auf 50.000 oder mehr v ausführenCPUs, sollten Sie die folgende Checkliste berücksichtigen.

> **Note**
>
> Wenn Sie planen, einen großen Workload mit einer Million oder mehr V CPUs auszuführen, oder wenn Sie Unterstützung bei der Ausführung eines großen Workloads benötigen, wenden Sie sich an Ihr AWS Team.

- Überprüfen Sie Ihre EC2 Amazon-Kontingente — Überprüfen Sie Ihre EC2 Amazon-Kontingente (auch als Limits bezeichnet) im Bereich Servicekontingente des AWS Management Console. Beantragen Sie bei Bedarf eine Erhöhung des Kontingents für Ihre maximale Anzahl an EC2 Amazon-Instances. Denken Sie daran, dass Amazon EC2 Spot- und Amazon On-Demand-Instances separate Kontingente haben. Weitere Informationen finden Sie unter Erste Schritte mit Service Quotas.
- Überprüfen Sie Ihr Amazon Elastic Block Store-Kontingent für jede Region — Jede Instance verwendet ein GP2 GP3 Oder-Volume für das Betriebssystem. Standardmäßig beträgt das Kontingent für jedes AWS-Region Objekt 300 TiB. Jede Instanz verwendet jedoch Zählungen als Teil dieses Kontingents. Denken Sie also daran, dies zu berücksichtigen, wenn Sie Ihr Amazon Elastic Block Store-Kontingent für jede Region verifizieren. Wenn Ihr Kontingent erreicht ist, können Sie keine weiteren Instances erstellen. Weitere Informationen finden Sie unter Amazon Elastic Block Store: Endpunkte und Kontingente
- Verwenden Sie Amazon S3 als Speicher — Amazon S3 bietet einen hohen Durchsatz und hilft, das Rätselraten darüber zu vermeiden, wie viel Speicherplatz basierend auf der Anzahl der Jobs und Instances in jeder Availability Zone bereitgestellt werden muss. Weitere Informationen finden Sie unter Bewährte Entwurfsmuster: Optimierung der Amazon S3 S3-Leistung.

- Skalieren Sie schrittweise, um Engpässe frühzeitig zu erkennen — Beginnen Sie bei einem Job, der mit einer Million oder mehr V ausgeführt wirdCPUs, niedriger und erhöhen Sie diese schrittweise, sodass Sie Engpässe frühzeitig erkennen können. Beginnen Sie beispielsweise mit der Ausführung auf 50.000 V. CPUs Erhöhen Sie dann die Anzahl auf 200 Tausend V CPUs und dann auf 500 Tausend V CPUs und so weiter. Mit anderen Worten, erhöhen Sie die Anzahl der vCPUs schrittweise weiter, bis Sie die gewünschte Anzahl von v CPUs erreicht haben.
- Überwachung, um potenzielle Probleme frühzeitig zu erkennen — Um mögliche Unterbrechungen und Probleme bei der Ausführung im großen Maßstab zu vermeiden, sollten Sie sowohl Ihre Anwendung als auch Ihre Architektur überwachen. Selbst bei einer Skalierung von 1.000 V auf 5.000 V können Unterbrechungen auftretenCPUs. Sie können Amazon CloudWatch Logs verwenden, um Protokolldaten zu überprüfen, oder CloudWatch Embedded Metrics mithilfe einer Client-Bibliothek verwenden. Weitere Informationen finden Sie unter CloudWatch Logs-Agent-Referenz und `aws-embedded-metrics`

Optimieren Sie Container und AMIs

Größe und Struktur des Containers sind wichtig für die erste Gruppe von Jobs, die Sie ausführen. Dies gilt insbesondere, wenn der Container größer als 4 GB ist. Container-Images werden in Schichten erstellt. Die Ebenen werden parallel von Docker mithilfe von drei gleichzeitigen Threads abgerufen. Mit dem Parameter können Sie die Anzahl der gleichzeitigen Threads erhöhen. `max-concurrent-downloads` Weitere Informationen finden Sie in der Dockerd-Dokumentation.

Sie können zwar größere Container verwenden, wir empfehlen jedoch, die Containerstruktur und -größe für schnellere Startzeiten zu optimieren.

- Kleinere Container werden schneller abgerufen — Kleinere Container können zu schnelleren Startzeiten von Anwendungen führen. Um die Containergröße zu verringern, laden Sie Bibliotheken oder Dateien, die selten aktualisiert werden, auf das Amazon Machine Image (AMI) aus. Sie können auch Bind-Mounts verwenden, um Zugriff auf Ihre Container zu gewähren. Weitere Informationen finden Sie unter Bind-Mounts.
- Erstellen Sie Ebenen mit gleichmäßiger Größe und teilen Sie große Schichten auf — Jede Ebene wird von einem Thread abgerufen. Eine große Ebene kann sich also erheblich auf die Startzeit Ihres Jobs auswirken. Wir empfehlen eine maximale Layer-Größe von 2 GB als guten Kompromiss zwischen einer größeren Containergröße und schnelleren Startzeiten. Sie können den `docker history your_image_id` Befehl ausführen, um die Struktur und die Layergröße Ihres Container-Images zu überprüfen. Weitere Informationen finden Sie in der Docker-Dokumentation.

- Verwenden Sie Amazon Elastic Container Registry als Ihr Container-Repository — Wenn Sie Tausende von Jobs parallel ausführen, kann ein selbstverwaltetes Repository ausfallen oder den Durchsatz drosseln. Amazon ECR ist skalierbar und kann Workloads mit bis zu über einer Million V bewältigen. CPUs

Wählen Sie die richtige Ressource für die Rechenumgebung

AWS Fargate erfordert weniger Ersteinrichtung und Konfiguration als Amazon EC2 und ist wahrscheinlich einfacher zu bedienen, insbesondere wenn es Ihr erstes Mal ist. Mit Fargate müssen Sie keine Server verwalten, Kapazitätsplanung durchführen oder Container-Workloads aus Sicherheitsgründen isolieren.

Wenn Sie die folgenden Anforderungen haben, empfehlen wir Ihnen, Fargate-Instances zu verwenden:

- Ihre Jobs müssen schnell beginnen, insbesondere in weniger als 30 Sekunden.
- Die Anforderungen Ihrer Jobs sind 16 V CPUs oder weniger GPUs, nein und 120 GiB Arbeitsspeicher oder weniger.

Weitere Informationen finden Sie unter Wann sollte Fargate verwendet werden.

Wenn Sie die folgenden Anforderungen haben, empfehlen wir Ihnen, EC2 Amazon-Instances zu verwenden:

- Sie benötigen mehr Kontrolle über die Instance-Auswahl oder möchten bestimmte Instance-Typen verwenden.
- Ihre Jobs benötigen Ressourcen, die nicht bereitgestellt werden AWS Fargate können, z. B. GPUs mehr Arbeitsspeicher, ein benutzerdefiniertes AMI oder den Amazon Elastic Fabric Adapter.
- Sie benötigen ein hohes Maß an Durchsatz oder Parallelität.
- Sie müssen Ihr AMI, Ihre Amazon EC2 Launch Template oder den Zugriff auf spezielle Linux-Parameter anpassen.

Mit Amazon EC2 können Sie Ihre Arbeitslast genauer auf Ihre spezifischen Anforderungen abstimmen und bei Bedarf skalieren.

Amazon EC2 On-Demand oder Amazon EC2 Spot

Die meisten AWS Batch Kunden verwenden Amazon EC2 Spot-Instances aufgrund der Einsparungen gegenüber On-Demand-Instances. Wenn Ihr Workload jedoch mehrere Stunden dauert und nicht unterbrochen werden kann, sind On-Demand-Instances möglicherweise besser für Sie geeignet. Sie können Spot-Instances immer zuerst testen und bei Bedarf zu On-Demand-Instances wechseln.

Wenn Sie die folgenden Anforderungen und Erwartungen haben, verwenden Sie Amazon EC2 On-Demand-Instances:

- Die Laufzeit Ihrer Jobs beträgt mehr als eine Stunde, und Sie können keine Unterbrechungen Ihrer Arbeitslast tolerieren.
- Sie haben ein striktes SLO (Service Level Objective) für Ihre gesamte Arbeitslast und können die Rechenzeit nicht verlängern.
- Bei den Instanzen, die Sie benötigen, ist es wahrscheinlicher, dass es zu Unterbrechungen kommt.

Wenn Sie die folgenden Anforderungen und Erwartungen haben, verwenden Sie Amazon EC2 Spot-Instances:

- Die Laufzeit Ihrer Jobs beträgt in der Regel 30 Minuten oder weniger.
- Sie können mögliche Unterbrechungen und Jobverschiebungen als Teil Ihrer Arbeitslast tolerieren. Weitere Informationen finden Sie unter Spot Instance Advisor.
- Jobs mit langer Laufzeit können von einem Checkpoint aus neu gestartet werden, wenn sie unterbrochen werden.

Sie können beide Kaufmodelle kombinieren, indem Sie zuerst auf der Spot-Instance einreichen und dann die On-Demand-Instance als Fallback-Option verwenden. Reichen Sie Ihre Jobs beispielsweise in einer Warteschlange ein, die mit Rechenumgebungen verbunden ist, die auf Amazon EC2 Spot-Instances ausgeführt werden. Wenn ein Job unterbrochen wird, catch Sie das Ereignis von Amazon ab EventBridge und korrelieren Sie es mit der Rückforderung einer Spot-Instance. Senden Sie den Job anschließend mithilfe einer Funktion oder erneut an eine On-Demand-Warteschlange weiter. AWS Lambda AWS Step Functions Weitere Informationen finden Sie unter Tutorial: Senden von Amazon Simple Notification Service-Benachrichtigungen für fehlgeschlagene Auftragsereignisse Bewährte Methoden für den Umgang mit Amazon EC2 Spot-Instance-Unterbrechungen und Manage AWS Batch with Step Functions.

> **Important**
>
> Verwenden Sie verschiedene Instance-Typen, Größen und Availability Zones für Ihre On-Demand-Datenverarbeitungsumgebung, um die Verfügbarkeit des Amazon EC2 Spot-Instance-Pools aufrechtzuerhalten und die Unterbrechungsrate zu verringern.

Verwenden Sie die Best Practices von Amazon EC2 Spot für AWS Batch

Wenn Sie sich für Amazon Elastic Compute Cloud (EC2) Spot-Instances entscheiden, können Sie Ihren Workflow wahrscheinlich optimieren, um Kosten zu sparen, manchmal sogar erheblich. Weitere Informationen finden Sie unter Bewährte Methoden für Amazon EC2 Spot.

Um Ihren Arbeitsablauf zu optimieren und Kosten zu sparen, sollten Sie die folgenden Best Practices von Amazon EC2 Spot in Betracht ziehen für AWS Batch:

- Wählen Sie die **`SPOT_CAPACITY_OPTIMIZED`** Zuweisungsstrategie — AWS Batch wählt EC2 Amazon-Instances aus den umfangreichsten Amazon EC2 Spot-Kapazitätspools aus. Wenn Sie sich Sorgen über Unterbrechungen machen, ist dies eine geeignete Wahl. Weitere Informationen finden Sie unter Strategien zur Zuweisung von Instance-Typen für AWS Batch.
- Instance-Typen diversifizieren — Um Ihre Instance-Typen zu diversifizieren, sollten Sie kompatible Größen und Familien in Betracht ziehen und dann je nach Preis oder Verfügbarkeit AWS Batch entscheiden. Betrachten Sie dies beispielsweise `c5.24xlarge` als Alternative zu `c5.12xlarge` oder`c5a`,, `c5n` `c5dm5`, und `m5d` Familien. Weitere Informationen finden Sie unter Seien Sie flexibel in Bezug auf Instanztypen und Availability Zones.

- Reduzieren Sie die Auftragslaufzeit oder den Checkpoint — Wir raten davon ab, Jobs auszuführen, die eine Stunde oder länger dauern, wenn Sie Amazon EC2 Spot-Instances verwenden, um Unterbrechungen zu vermeiden. Wenn Sie Ihre Jobs in kleinere Teile, die aus 30 Minuten oder weniger bestehen, aufteilen oder Checkpoints unterteilen, können Sie die Wahrscheinlichkeit von Unterbrechungen erheblich reduzieren.
- Automatisierte Wiederholungen verwenden — Um Unterbrechungen bei Aufträgen zu vermeiden, sollten Sie automatische Wiederholungen für AWS Batch Jobs einrichten. Batch-Jobs können aus einem der folgenden Gründe unterbrochen werden: Es wird ein Exit-Code ungleich Null zurückgegeben, es tritt ein Servicefehler auf oder es erfolgt eine Instanzrückforderung. Sie können bis zu 10 automatische Wiederholungsversuche einrichten. Zunächst empfehlen wir, dass Sie mindestens 1—3 automatische Wiederholungen einrichten. Informationen zur Nachverfolgung von Amazon EC2 Spot-Unterbrechungen finden Sie unter Spot-Unterbrechungs-Dashboard.

  Denn AWS Batch wenn Sie den Wiederholungsparameter festlegen, wird der Job ganz vorne in der Job-Warteschlange platziert. Das heißt, dem Job wird Priorität eingeräumt. Wenn Sie die Jobdefinition erstellen oder den Job in der einreichen AWS CLI, können Sie eine Wiederholungsstrategie konfigurieren. Weitere Informationen finden Sie unter submit-job.

  ```
  $ aws batch submit-job --job-name MyJob \
      --job-queue MyJQ \
      --job-definition MyJD \
      --retry-strategy attempts=2
  ```

- Verwenden Sie benutzerdefinierte Wiederholungsversuche — Sie können eine Strategie für die Wiederholung von Aufträgen für einen bestimmten Anwendungs-Exit-Code oder die Rückforderung einer Instanz konfigurieren. Im folgenden Beispiel kann der Job bis zu fünf Mal wiederholt werden, wenn der Host den Fehler verursacht. Wenn der Job jedoch aus einem anderen Grund fehlschlägt, wird der Job beendet und der Status wird auf gesetzt. `FAILED`

  ```
  "retryStrategy": {
      "attempts": 5,
      "evaluateOnExit":
      [{
          "onStatusReason" :"Host EC2*",
          "action": "RETRY"
      },{
        "onReason" : "*",
          "action": "EXIT"
      }]
  ```

```
}
```

- Verwenden Sie das Spot-Interruption-Dashboard — Sie können das Spot-Interruption-Dashboard verwenden, um Spot-Unterbrechungen nachzuverfolgen. Die Anwendung bietet Metriken zu zurückgeforderten Amazon EC2 Spot-Instances und zu den Availability Zones, in denen sich Spot-Instances befinden. Weitere Informationen finden Sie unter Spot-Interruption-Dashboard

Häufige Fehler und Problembehebung

Fehler treten AWS Batch häufig auf Anwendungsebene auf oder werden durch Instanzkonfigurationen verursacht, die Ihren spezifischen Jobanforderungen nicht entsprechen. Zu den weiteren Problemen gehören Jobs, die im `RUNNABLE` Status hängen bleiben, oder Rechenumgebungen, die in einem `INVALID` Status stecken bleiben. Weitere Informationen zur Fehlerbehebung bei Jobs, die im `RUNNABLE` Status hängen bleiben, finden Sie unterJobs, die in einem `RUNNABLE` Status feststecken. Informationen zur Fehlerbehebung bei Computerumgebungen in einem bestimmten `INVALID` Bundesstaat finden Sie unter`INVALID`Datenverarbeitungsumgebung.

- Überprüfen Sie die vCPU-Kontingente von Amazon EC2 Spot — Stellen Sie sicher, dass Ihre aktuellen Service-Kontingente den Auftragsanforderungen entsprechen. Nehmen wir zum Beispiel an, dass Ihr aktuelles Servicekontingent 256 V beträgt CPUs und der Job 10.000 V CPUs erfordert. Dann entspricht das Servicekontingent nicht den Auftragsanforderungen. Weitere Informationen und Anweisungen zur Fehlerbehebung finden Sie unter EC2 Amazon-Servicekontingente und Wie erhöhe ich das Servicekontingent meines Amazon EC2resources? .
- Jobs schlagen fehl, bevor die Anwendung ausgeführt wird — Einige Jobs schlagen möglicherweise aufgrund eines `DockerTimeoutError` Fehlers oder eines `CannotPullContainerError` Fehlers fehl. Informationen zur Problembehandlung finden Sie unter Wie behebe ich den Fehler DockerTimeoutError "" in AWS Batch? .
- Unzureichende IP-Adressen — Die Anzahl der IP-Adressen in Ihrer VPC und Ihren Subnetzen kann die Anzahl der Instances einschränken, die Sie erstellen können. Verwenden Sie Classless Inter-Domain Routings (CIDRs), um mehr IP-Adressen bereitzustellen, als für die Ausführung Ihrer Workloads erforderlich sind. Bei Bedarf können Sie auch eine dedizierte VPC mit einem großen Adressraum erstellen. Sie können beispielsweise eine VPC mit mehreren Eingängen `10.x.0.0/16` und ein Subnetz CIDRs in jeder Availability Zone mit einem CIDR von erstellen. `10.x.y.0/17` In diesem Beispiel liegt x zwischen 1 und 4 und y ist entweder 0 oder 128. Diese Konfiguration bietet 36.000 IP-Adressen in jedem Subnetz.

- Stellen Sie sicher, dass Instances bei Amazon registriert sind EC2 — Wenn Sie Ihre Instances in der EC2 Amazon-Konsole sehen, aber keine Amazon Elastic Container Service-Container-Instances in Ihrem Amazon ECS-Cluster, ist der Amazon ECS-Agent möglicherweise nicht auf einem Amazon Machine Image (AMI) installiert. Der Amazon ECS-Agent, die EC2 Amazon-Daten in Ihrem AMI oder die Startvorlage sind möglicherweise ebenfalls nicht richtig konfiguriert. Um die Ursache zu isolieren, erstellen Sie eine separate EC2 Amazon-Instance oder stellen Sie über SSH eine Verbindung zu einer vorhandenen Instance her. Weitere Informationen finden Sie unter Konfiguration des Amazon ECS-Container-Agenten, Speicherorte von Amazon ECS-Protokolldateien undRessource berechnen AMIs.

- Überprüfen Sie das AWS Dashboard — Überprüfen Sie das AWS Dashboard, um zu überprüfen, ob der erwartete Auftragsstatus vorliegt und ob die Rechenumgebung erwartungsgemäß skaliert wird. Sie können auch die Job-Logs überprüfen CloudWatch.
- Stellen Sie sicher, dass Ihre Instanz erstellt wurde — Wenn eine Instanz erstellt wurde, bedeutet dies, dass Ihre Computerumgebung wie erwartet skaliert wurde. Wenn Ihre Instanzen nicht erstellt wurden, suchen Sie nach den zugehörigen Subnetzen in Ihrer Computerumgebung, die Sie ändern möchten. Weitere Informationen finden Sie unter Überprüfen einer Skalierungsaktivität für eine Auto Scaling Scaling-Gruppe.

  Wir empfehlen Ihnen außerdem, zu überprüfen, ob Ihre Instances Ihre entsprechenden Jobanforderungen erfüllen können. Beispielsweise kann ein Job 1 TiB Arbeitsspeicher benötigen, aber die Rechenumgebung verwendet einen C5-Instance-Typ, der auf 192 GB Speicher begrenzt ist.
- Stellen Sie sicher, dass Ihre Instances angefordert werden von AWS Batch — Überprüfen Sie den Auto Scaling Scaling-Gruppenverlauf, um zu überprüfen, ob Ihre Instances von angefordert wurden AWS Batch. Dies ist ein Hinweis darauf, wie Amazon EC2 versucht, Instances zu erwerben. Wenn Sie eine Fehlermeldung erhalten, dass Amazon EC2 Spot keine Instance in einer bestimmten Availability Zone erwerben kann, liegt das möglicherweise daran, dass die Availability Zone keine bestimmte Instance-Familie anbietet.
- Stellen Sie sicher, dass Instances bei Amazon ECS registriert sind — Wenn Sie Instances in der EC2 Amazon-Konsole sehen, aber keine Amazon ECS-Container-Instances in Ihrem Amazon ECS-Cluster, ist der Amazon ECS-Agent möglicherweise nicht auf dem Amazon Machine Image (AMI) installiert. Darüber hinaus sind der Amazon ECS-Agent, die EC2 Amazon-Daten in Ihrem AMI oder die Startvorlage möglicherweise nicht richtig konfiguriert. Um die Ursache zu isolieren, erstellen Sie eine separate EC2 Amazon-Instance oder stellen Sie über SSH eine Verbindung zu einer vorhandenen Instance her. Weitere Informationen finden Sie in der CloudWatch Agenten-Konfigurationsdatei: Abschnitt Protokolle, Speicherorte der Amazon ECS-Protokolldateien undRessource berechnen AMIs.
- Öffnen Sie ein Support-Ticket — Wenn Sie nach der Fehlerbehebung immer noch Probleme haben und über einen Supportplan verfügen, öffnen Sie ein Support-Ticket. Stellen Sie sicher, dass das Support-Ticket Informationen zum Problem, zu den Besonderheiten der Arbeitslast, zur Konfiguration und zu den Testergebnissen enthält. Weitere Informationen finden Sie unter Support Tarife vergleichen.
- Weitere Informationen finden Sie in den Foren AWS Batch und HPC. Weitere Informationen finden Sie in den Foren AWS Batchund HPC.

- Sehen Sie sich das AWS Batch Runtime Monitoring Dashboard an — Dieses Dashboard verwendet eine serverlose Architektur, um Ereignisse von Amazon ECS und Amazon EC2 zu erfassen AWS Batch, um Einblicke in Jobs und Instances zu erhalten. Weitere Informationen finden Sie unter AWS Batch Runtime Monitoring Dashboards Solution.

Problembehebung AWS Batch

Möglicherweise müssen Sie Probleme beheben, die mit Ihren Rechenumgebungen, Job-Warteschlangen, Jobdefinitionen oder Jobs zusammenhängen. In diesem Kapitel wird beschrieben, wie Sie solche Probleme in Ihrer AWS Batch Umgebung beheben und lösen können.

AWS Batch verwendet IAM-Richtlinien, -Rollen und -Berechtigungen und läuft auf der Infrastruktur von Amazon EC2, Amazon ECS und Amazon Elastic Kubernetes Service. AWS Fargate Informationen zur Behebung von Problemen im Zusammenhang mit diesen Diensten finden Sie im Folgenden:

- Problembehandlung bei IAM im IAM-Benutzerhandbuch
- Amazon ECS-Fehlerbehebung im Amazon Elastic Container Service Developer Guide
- Amazon EKS-Fehlerbehebung im Amazon EKS-Benutzerhandbuch
- Fehlerbehebung bei EC2 Instances im EC2 Amazon-Benutzerhandbuch

Inhalt

- AWS Batch
  - INVALIDDatenverarbeitungsumgebung
    - Falscher Rollenname oder ARN
    - Reparieren Sie eine INVALID Computerumgebung
  - Jobs, die in einem RUNNABLE Status feststecken
  - Spot-Instances wurden bei der Erstellung nicht markiert
  - Spot-Instances werden nicht herunterskaliert
    - Fügen Sie Ihrer Spot-Flottenrolle eine von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie hinzu in der AWS Management Console
    - Fügen Sie Ihrer Spot-Flottenrolle eine von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie hinzu mit dem AWS CLI
  - Secrets Manager können nicht abgerufen werden
  - Die Ressourcenanforderungen für die Jobdefinition können nicht außer Kraft gesetzt werden
  - Fehlermeldung beim Aktualisieren der desiredvCpus Einstellung
- AWS Batch auf Amazon EKS
  - INVALIDRechenumgebung

    - Nicht unterstützt Kubernetes version
    - Instanzprofil existiert nicht
    - Ungültig Kubernetes Namespace
    - Computerumgebung gelöscht
    - Knoten treten dem Amazon EKS-Cluster nicht bei
- AWS Batch bei Amazon EKS bleibt der Job im RUNNABLE Status hängen
- AWS Batch bei Amazon EKS bleibt der Job im STARTING Status hängen
    - Szenario: Dauerhafter Fehler beim Anhängen oder Bereitstellen von Volumenansprüchen
- Stellen Sie sicher, dass der richtig konfiguriert aws-auth ConfigMap ist
- RBAC-Berechtigungen oder -Bindungen sind nicht richtig konfiguriert

AWS Batch

Lesen Sie sich die folgenden Themen durch, um Überprüfungsverfahren und mögliche Lösungen für häufig auftretende Probleme zu finden, die bei der Verwendung auftreten können AWS Batch.

Themen

- INVALIDDatenverarbeitungsumgebung
- Jobs, die in einem RUNNABLE Status feststecken
- Spot-Instances wurden bei der Erstellung nicht markiert
- Spot-Instances werden nicht herunterskaliert
- Secrets Manager können nicht abgerufen werden
- Die Ressourcenanforderungen für die Jobdefinition können nicht außer Kraft gesetzt werden
- Fehlermeldung beim Aktualisieren der desiredvCpus Einstellung

`INVALID`Datenverarbeitungsumgebung

Es ist möglich, dass Sie eine verwaltete Rechenumgebung falsch konfiguriert haben. Wenn Sie das getan haben, wechselt die Computerumgebung in einen `INVALID` Status und kann keine Stellen zur Vermittlung annehmen. In den folgenden Abschnitten werden die möglichen Ursachen beschrieben und wie Sie anhand der Ursache Fehler beheben können.

> **Important**
>
> AWS Batch erstellt und verwaltet mehrere AWS Ressourcen in Ihrem Namen und innerhalb Ihres Kontos, darunter Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets und Amazon ECS-Cluster. Diese verwalteten Ressourcen sind speziell konfiguriert, um einen optimalen AWS Batch Betrieb zu gewährleisten. Die manuelle Änderung dieser im Batch verwalteten Ressourcen kann, sofern nicht ausdrücklich in der AWS Batch Dokumentation angegeben, zu unerwartetem Verhalten führen, das zu einer `INVALID` Rechenumgebung, einem suboptimalen Skalierungsverhalten der Instanz, verzögerter Workload-Verarbeitung oder unerwarteten Kosten führen kann. Diese manuellen Änderungen können vom Service nicht deterministisch unterstützt werden. AWS Batch Verwenden Sie immer die unterstützte Batch APIs - oder Batch-Konsole, um Ihre Rechenumgebungen zu verwalten.

Falscher Rollenname oder ARN

Der häufigste Grund dafür, dass eine Rechenumgebung in einen `INVALID` Status wechselt, ist, dass die AWS Batch Servicerolle oder die Amazon EC2 Spot-Flottenrolle einen falschen Namen oder Amazon Resource Name (ARN) hat. Dies ist häufiger bei Rechenumgebungen der Fall, die mit AWS CLI oder dem erstellt wurden AWS SDKs. Wenn Sie eine Rechenumgebung in der erstellen AWS Management Console, AWS Batch hilft Ihnen bei der Auswahl der richtigen Service- oder Spot-Flottenrollen. Nehmen wir jedoch an, Sie geben den Namen oder den ARN manuell ein und geben sie falsch ein. Dann ist es auch die resultierende Rechenumgebung`INVALID`.

Nehmen wir jedoch an, dass Sie den Namen oder den ARN für eine IAM-Ressource manuell in einen AWS CLI Befehl oder Ihren SDK-Code eingeben. In diesem Fall AWS Batch kann die Zeichenfolge nicht validiert werden. Stattdessen AWS Batch müssen Sie den falschen Wert akzeptieren und versuchen, die Umgebung zu erstellen. Wenn AWS Batch die Umgebung nicht erstellt werden kann, wechselt die Umgebung in einen `INVALID` Status, und es werden die folgenden Fehler angezeigt.

Bei einer ungültigen Servicerolle:

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

Bei einer ungültigen Spot-Flottenrolle:

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

Eine häufige Ursache für dieses Problem ist das folgende Szenario. Sie geben nur den Namen einer IAM-Rolle an AWS SDKs, wenn Sie den AWS CLI oder den anstelle des vollständigen Amazon-Ressourcennamens (ARN) verwenden. Je nachdem, wie Sie die Rolle erstellt haben, kann der ARN ein `aws-service-role` Pfadpräfix enthalten. Wenn Sie die AWS Batch Servicerolle beispielsweise mithilfe der Verfahren unter manuell erstellen[Verwenden Sie dienstbezogene Rollen für AWS Batch](), könnte Ihr ARN für die Servicerolle wie folgt aussehen.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Wenn Sie die Servicerolle heute jedoch als Teil des Assistenten für die erste Ausführung der Konsole erstellt haben, könnte Ihr ARN für die Servicerolle wie folgt aussehen.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Dieses Problem kann auch auftreten, wenn Sie die AWS Batch Service-Level-Richtlinie (`AWSBatchServiceRole`) an eine Rolle anhängen, die nichts mit dem Dienst zu tun hat. In diesem Szenario erhalten Sie beispielsweise möglicherweise eine Fehlermeldung, die der folgenden ähnelt:

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
     authorized to perform: action on resource ...
```

Gehen Sie wie folgt vor, um dieses Problem zu beheben.

- Verwenden Sie eine leere Zeichenfolge für die Servicerolle, wenn Sie die AWS Batch Rechenumgebung erstellen.
- Geben Sie die Servicerolle im folgenden Format an:`arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Wenn Sie nur den Namen einer IAM-Rolle angeben, wenn Sie das AWS CLI oder das verwenden AWS SDKs, AWS Batch wird davon ausgegangen, dass Ihr ARN das `aws-service-role`

Pfadpräfix nicht verwendet. Aus diesem Grund empfehlen wir, dass Sie bei der Erstellung von Rechenumgebungen den vollständigen ARN für Ihre IAM-Rollen angeben.

Informationen zum Reparieren einer Computerumgebung, die auf diese Weise falsch konfiguriert ist, finden Sie unter. Reparieren Sie eine `INVALID` Computerumgebung

Reparieren Sie eine **`INVALID`** Computerumgebung

Wenn sich eine Rechenumgebung in einem `INVALID` Zustand befindet, aktualisieren Sie sie, um den ungültigen Parameter zu reparieren. Aktualisieren Sie in einem Falscher Rollenname oder ARN Fall die Rechenumgebung mit der richtigen Servicerolle.

So Reparieren Sie eine falsch konfigurierte Datenverarbeitungsumgebung

1. Öffnen Sie die AWS Batch Konsole unter https://console.aws.amazon.com/batch/.
2. Wählen Sie in der Navigationsleiste die aus, die Sie verwenden AWS-Region möchten.
3. Wählen Sie im Navigationsbereich Datenverarbeitungs-Umgebungen aus.
4. Wählen Sie auf der Seite Datenverarbeitungs-Umgebungen das Optionsfeld neben der zu bearbeitenden Datenverarbeitungsumgebung und dann Bearbeiten aus.
5. Wählen Sie auf der Seite Rechenumgebung aktualisieren unter Servicerolle die IAM-Rolle aus, die Sie mit Ihrer Rechenumgebung verwenden möchten. Die AWS Batch -Konsole zeigt nur mit der richtigen Vertrauensstellung für Datenverarbeitungsumgebungen an.
6. Wählen Sie Speichern aus, um Ihre Datenverarbeitungsumgebung zu aktualisieren.

Jobs, die in einem **`RUNNABLE`** Status feststecken

Angenommen, Ihre Datenverarbeitungsumgebung enthält Rechenressourcen, aber Ihre Jobs werden nicht über diesen `RUNNABLE` Status hinaus bearbeitet. Dann ist es wahrscheinlich, dass etwas verhindert, dass die Jobs auf einer Rechenressource platziert werden, und dass Ihre Jobwarteschlangen blockiert werden. Hier erfahren Sie, ob Ihr Job darauf wartet, an der Reihe zu sein, oder ob er feststeckt und die Warteschlange blockiert.

Wenn AWS Batch festgestellt wird, dass Sie einen `RUNNABLE` Job an der Spitze haben und die Warteschlange blockieren, erhalten Sie von Amazon CloudWatch Events eine Ressource: Blockierte Ereignisse in der Jobwarteschlange Veranstaltung mit dem Grund. Derselbe Grund wird auch im Rahmen von `ListJobs` `DescribeJobs` API-Aufrufen in das `statusReason` Feld aktualisiert.

Optional können Sie den `jobStateTimeLimitActions` Parameter über `CreateJobQueue` und `UpdateJobQueue`API-Aktionen konfigurieren.

> **Note**
>
> Derzeit besteht die einzige Aktion, die Sie mit verwenden können, `jobStateLimitActions.action` darin, einen Job abzubrechen.

Der `jobStateTimeLimitActions` Parameter wird verwendet, um eine Reihe von Aktionen anzugeben, die AWS Batch für Jobs in einem bestimmten Status ausgeführt werden. Sie können über das `maxTimeSeconds` Feld einen Zeitschwellenwert in Sekunden festlegen.

Wenn sich ein Job in einem `RUNNABLE` Zustand mit dem definierten Status befunden hat `statusReason` AWS Batch , wird die angegebene Aktion nach Ablauf `maxTimeSeconds` ausgeführt.

Sie können den `jobStateTimeLimitActions` Parameter beispielsweise so einstellen, dass bis zu 4 Stunden auf jeden Job in diesem `RUNNABLE` Status gewartet wird, der darauf wartet, dass genügend Kapazität verfügbar wird. Sie können dies tun, indem Sie `statusReason` den `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY` Wert `maxTimeSeconds` auf 144000 einstellen, bevor Sie den Job abbrechen und dem nächsten Job erlauben, an die Spitze der Job-Warteschlange zu gelangen.

Im Folgenden sind die Gründe aufgeführt, aus denen AWS Batch hervorgeht, dass eine Auftragswarteschlange blockiert ist. Diese Liste enthält die Nachrichten, die von den Aktionen `ListJobs` und der `DescribeJobs` API zurückgegeben wurden. Dies sind auch dieselben Werte, die Sie für den `jobStateLimitActions.statusReason` Parameter definieren können.

1. Grund: In allen verbundenen Rechenumgebungen treten Fehler aufgrund unzureichender Kapazitäten auf. AWS Batch Erkennt auf Anfrage EC2 Amazon-Instances, bei denen Fehler mit unzureichender Kapazität auftreten. Wenn Sie den Auftrag manuell stornieren, wird der nachfolgende Auftrag an die Spitze der Warteschlange verschoben. Ohne die Probleme mit der Servicerolle zu lösen, ist es jedoch wahrscheinlich, dass auch der nächste Auftrag blockiert wird. Es empfiehlt sich, dieses Problem manuell zu untersuchen und zu lösen.
   - **`statusReason`**Nachricht, während der Job feststeckt: `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]`

- **`reason`**verwendet für**`jobStateTimeLimitActions`**:
  `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`
- **`statusReason`**Nachricht, nachdem der Job abgebrochen wurde:
  `Canceled by JobStateTimeLimit action due to reason: CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`

Hinweis:

a. Die AWS Batch Servicerolle benötigt eine `autoscaling:DescribeScalingActivities` entsprechende Genehmigung, damit diese Erkennung funktioniert. Wenn Sie die Dienstbezogene Rollenberechtigungen für AWS Batch serviceverknüpfte Rolle (SLR) oder die AWS verwaltete Richtlinie: AWSBatchServiceRoleRichtlinie verwaltete Richtlinie verwenden, müssen Sie keine Maßnahmen ergreifen, da die entsprechenden Berechtigungsrichtlinien aktualisiert wurden.

b. Wenn Sie die SLR- oder die verwaltete Richtlinie verwenden, müssen Sie die `ec2:DescribeSpotFleetRequestHistory` Berechtigungen `autoscaling:DescribeScalingActivities` und hinzufügen, damit Sie blockierte Jobwarteschlangenereignisse und den aktualisierten Auftragsstatus erhalten können, wenn Sie angemeldet sind. `RUNNABLE` Darüber hinaus sind diese AWS Batch Berechtigungen erforderlich, um `cancellation` Aktionen über den `jobStateTimeLimitActions` Parameter ausführen zu können, auch wenn sie in der Auftragswarteschlange konfiguriert sind.

c. Wenn bei einem MNP-Job (Multi-Node Parallel) Fehler in der angehängten EC2 Amazon-Rechenumgebung mit hoher Priorität auftreten, wird die Warteschlange blockiert, selbst wenn dieser `insufficient capacity` Fehler in einer Rechenumgebung mit niedrigerer Priorität auftritt.

2. Grund: Alle Computerumgebungen haben einen `maxvCpus`Parameter, der kleiner ist als die Jobanforderungen. Wenn Sie den Job entweder manuell oder durch Aktivieren des `jobStateTimeLimitActions` Parameters abbrechen`statusReason`, kann der nachfolgende Job an die Spitze der Warteschlange verschoben werden. Optional können Sie den `maxvCpus` Parameter der primären Rechenumgebung erhöhen, um den Anforderungen des blockierten Auftrags gerecht zu werden.

   - **`statusReason`**Nachricht, während der Job feststeckt:
     `MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.`
   - **`reason`**verwendet für**`jobStateTimeLimitActions`**:
     `MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`

- **statusReason**Nachricht, nachdem der Job abgebrochen wurde:
  `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`

3. Grund: Keine der Rechenumgebungen verfügt über Instanzen, die die Jobanforderungen erfüllen. Wenn ein Job Ressourcen anfordert, wird AWS Batch erkannt, dass keine angeschlossene Rechenumgebung in der Lage ist, den eingehenden Job zu verarbeiten. Wenn Sie den Job entweder manuell oder durch Aktivieren des `jobStateTimeLimitActions` Parameters abbrechen`statusReason`, kann der nachfolgende Job an die Spitze der Warteschlange verschoben werden. Optional können Sie die zulässigen Instanztypen der Rechenumgebung neu definieren, um die erforderlichen Jobressourcen hinzuzufügen.
   - **statusReason**Nachricht, während der Job feststeckt:
     `MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.`
   - **reason**verwendet für**jobStateTimeLimitActions**:
     `MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
   - **statusReason**Nachricht, nachdem der Job abgebrochen wurde:
     `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
4. Grund: In allen Computerumgebungen gibt es Probleme mit Servicerollen. Um dieses Problem zu lösen, vergleichen Sie Ihre Berechtigungen für Servicerollen mit AWS verwaltete Richtlinien für AWS Batch denen und schließen Sie etwaige Lücken. Hinweis: Sie können über den `jobStateTimeLimitActions` Parameter keine programmierbare Aktion konfigurieren, um diesen Fehler zu beheben.

   Es hat sich bewährt, den zu verwenden, um ähnliche Dienstbezogene Rollenberechtigungen für AWS Batch Fehler zu vermeiden.

   Wenn Sie den Auftrag entweder manuell oder durch Aktivieren des `jobStateTimeLimitActions` Parameters abbrechen`statusReason`, kann der nachfolgende Auftrag an die Spitze der Warteschlange verschoben werden. Ohne die Probleme mit der Servicerolle zu lösen, ist es wahrscheinlich, dass auch der nächste Auftrag blockiert wird. Es empfiehlt sich, dieses Problem manuell zu untersuchen und zu lösen.
   - **statusReason**Nachricht, während der Job feststeckt:
     `MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS – Batch service role has a permission issue.`

5. Grund: Alle Rechenumgebungen sind ungültig. Weitere Informationen finden Sie unter `INVALID`Datenverarbeitungsumgebung. Hinweis: Sie können über den `jobStateTimeLimitActions` Parameter keine programmierbare Aktion konfigurieren, um diesen Fehler zu beheben.
   - **`statusReason`**Meldung, während der Job feststeckt: `ACTION_REQUIRED - CE(s) associated with the job queue are invalid.`
6. Grund: AWS Batch hat eine blockierte Warteschlange erkannt, kann aber den Grund nicht ermitteln. Hinweis: Sie können über den `jobStateTimeLimitActions` Parameter keine programmierbare Aktion konfigurieren, um diesen Fehler zu beheben. Weitere Informationen zur Fehlerbehebung finden Sie unter Warum ist mein AWS Batch Job in RUNNABLE hängen AWS geblieben in re:POST.
   - **`statusReason`**Nachricht, während der Job feststeckt: `UNDETERMINED - Batch job is blocked, root cause is undetermined.`

Falls Sie kein Ereignis von CloudWatch Events oder ein Ereignis aus unbekanntem Grund erhalten haben, finden Sie hier einige der häufigsten Ursachen für dieses Problem.

Der **`awslogs`** Protokolltreiber ist auf Ihren Rechenressourcen nicht konfiguriert

AWS Batch Jobs senden ihre Protokollinformationen an CloudWatch Logs. Um die entsprechende Funktion zu aktivieren, müssen Sie Ihre Datenverarbeitungsressourcen so konfigurieren, dass sie den `awslogs`-Protokolltreiber verwenden. Angenommen, Sie basieren Ihr Computerressourcen-AMI auf dem für Amazon ECS optimierten AMI (oder Amazon Linux). Dann ist dieser Treiber standardmäßig im `ecs-init` Paket registriert. Nehmen wir nun an, Sie verwenden ein anderes Basis-AMI. Anschließend müssen Sie überprüfen, ob der `awslogs` Protokolltreiber als verfügbarer Protokolltreiber mit der `ECS_AVAILABLE_LOGGING_DRIVERS` Umgebungsvariablen angegeben ist, wenn der Amazon ECS-Container-Agent gestartet wird. Weitere Informationen erhalten Sie unter AMI-Spezifikation für Rechenressourcen und Tutorial: Erstellen Sie ein Compute-Ressourcen-AMI.

Unzureichende Ressourcen

Wenn in Ihren Jobdefinitionen mehr CPU- oder Speicherressourcen angegeben sind, als Ihre Rechenressourcen zuweisen können, werden Ihre Jobs niemals vergeben. Nehmen wir zum Beispiel an, dass Ihr Job 4 GiB Arbeitsspeicher spezifiziert und für Ihre Rechenressourcen weniger als der verfügbare Speicherplatz zur Verfügung steht. Dann ist es so, dass der Job nicht auf diesen Rechenressourcen platziert werden kann. In diesem Fall müssen Sie den angegebenen Speicher in Ihrer Auftragsdefinition reduzieren oder Ihrer Umgebung größere

Datenverarbeitungsressourcen hinzufügen. Ein Teil des Speichers ist für den Amazon ECS-Container-Agenten und andere kritische Systemprozesse reserviert. Weitere Informationen finden Sie unter Speicherverwaltung für Rechenressourcen.

Kein Internetzugang für Rechenressourcen

Datenverarbeitungsressourcen benötigen einen externen Netzwerkzugriff, um mit dem Amazon-ECS-Service-Endpunkt zu kommunizieren. Dies kann über einen Schnittstellen-VPC-Endpunkt oder über Ihre Datenverarbeitungsressourcen mit öffentlichen IP-Adressen sein.

Weitere Informationen zu Interface-VPC-Endpunkten finden Sie unter Amazon-ECS-Schnittstellen-VPC-Endpunkte (AWS PrivateLink) im Amazon Elastic Container Service-Entwicklerhandbuch.

Wenn Sie keinen Schnittstellen-VPC-Endpunkt konfiguriert haben und Ihre Datenverarbeitungsressourcen keine öffentlichen IP-Adressen haben, müssen Sie Netzwerkadressübersetzung (Network Address Translation, NAT) verwenden, um diesen Zugriff zur Verfügung zu stellen. Weitere Informationen finden Sie unter NAT-Gateways im Amazon VPC-Benutzerhandbuch. Weitere Informationen finden Sie unter the section called “Erstellen einer VPC”.

EC2 Amazon-Instanzlimit erreicht

Die Anzahl der EC2 Amazon-Instances, in denen Ihr Konto gestartet werden kann, hängt von Ihrem EC2 Instance-Kontingent ab. AWS-Region Bestimmte Instance-Typen haben auch ein per-instance-type Kontingent. Weitere Informationen zum EC2 Amazon-Instance-Kontingent Ihres Kontos, einschließlich der Beantragung einer Limiterhöhung, finden Sie unter Amazon EC2 Service Limits im EC2 Amazon-Benutzerhandbuch.

Der Amazon ECS-Container-Agent ist nicht installiert

Der Amazon ECS-Container-Agent muss auf dem Amazon Machine Image (AMI) installiert sein, damit Jobs AWS Batch ausgeführt werden können. Der Amazon ECS-Container-Agent ist standardmäßig auf Amazon ECS Optimized installiert AMIs. Weitere Informationen zum Amazon ECS-Container-Agenten finden Sie unter Amazon ECS-Container-Agent im Amazon Elastic Container Service Developer Guide.

Weitere Informationen finden Sie unter Warum bleibt mein AWS Batch Job im `RUNNABLE` Status hängen? in re:POST.

Spot-Instances wurden bei der Erstellung nicht markiert

Spot-Instance-Tagging für AWS Batch Rechenressourcen wird ab dem 25. Oktober 2017 unterstützt. Zuvor enthielt die empfohlene IAM-verwaltete Richtlinie (`AmazonEC2SpotFleetRole`) für die Amazon EC2 Spot-Flottenrolle keine Berechtigungen zum Taggen von Spot-Instances beim Start. Die neue empfohlene IAM-verwaltete Richtlinie heißt. `AmazonEC2SpotFleetTaggingRole` Sie unterstützt das Taggen von Spot-Instances beim Start.

Um das Spot-Instance-Tagging bei der Erstellung zu korrigieren, gehen Sie wie folgt vor, um die aktuell empfohlene IAM-verwaltete Richtlinie auf Ihre Amazon EC2 Spot-Flottenrolle anzuwenden. Auf diese Weise sind alle future Spot-Instances, die mit dieser Rolle erstellt werden, berechtigt, Instance-Tags anzuwenden, wenn sie erstellt werden.

So wenden Sie die aktuelle IAM-verwaltete Richtlinie auf Ihre Amazon EC2 Spot-Flottenrolle an

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie Rollen und anschließend Ihre Amazon EC2 Spot-Flotte-Rolle aus.
3. Wählen Sie Richtlinie anfügen aus.
4. Wählen Sie Amazon EC2 SpotFleetTaggingRole und dann Attach policy aus.
5. Wählen Sie erneut Ihre Amazon EC2 Spot-Flottenrolle aus, um die vorherige Richtlinie zu entfernen.
6. Wählen Sie das X rechts neben der EC2SpotFleetRoleAmazon-Richtlinie aus und wählen Sie Trennen aus.

Spot-Instances werden nicht herunterskaliert

AWS Batch hat am 10. März 2021 die AWSServiceRoleForBatchserviceverknüpfte Rolle eingeführt. Wenn im `serviceRole` Parameter der Rechenumgebung keine Rolle angegeben ist, wird diese dienstverknüpfte Rolle als Servicerolle verwendet. Nehmen wir jedoch an, dass die serviceverknüpfte Rolle in einer EC2 Spot-Computing-Umgebung verwendet wird, die verwendete Spot-Rolle jedoch nicht die von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie beinhaltet. Dann wird die Spot-Instance nicht herunterskaliert. Infolgedessen erhalten Sie eine Fehlermeldung mit der folgenden Meldung: „Sie sind nicht berechtigt, diesen Vorgang auszuführen.“ Gehen Sie wie folgt vor, um die Spot-Flottenrolle zu aktualisieren, die Sie im `spotIamFleetRole` Parameter verwenden. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen und Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Service im IAM-Benutzerhandbuch.

Themen

- Fügen Sie Ihrer Spot-Flottenrolle eine von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie hinzu in der AWS Management Console
- Fügen Sie Ihrer Spot-Flottenrolle eine von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie hinzu mit dem AWS CLI

Fügen Sie Ihrer Spot-Flottenrolle eine von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie hinzu in der AWS Management Console

So wenden Sie die aktuelle IAM-verwaltete Richtlinie auf Ihre Amazon EC2 Spot-Flottenrolle an

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie Rollen und anschließend Ihre Amazon EC2 Spot-Flotte-Rolle aus.
3. Wählen Sie Richtlinie anfügen aus.
4. Wählen Sie Amazon EC2 SpotFleetTaggingRole und dann Attach policy aus.
5. Wählen Sie erneut Ihre Amazon EC2 Spot-Flottenrolle aus, um die vorherige Richtlinie zu entfernen.
6. Wählen Sie das X rechts neben der EC2SpotFleetRoleAmazon-Richtlinie aus und wählen Sie Trennen aus.

Fügen Sie Ihrer Spot-Flottenrolle eine von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie hinzu mit dem AWS CLI

Bei den Beispielbefehlen wird davon ausgegangen, dass Ihre Amazon EC2 Spot-Flottenrolle benannt ist*`AmazonEC2SpotFleetRole`*. Wenn Ihre Rolle einen anderen Namen verwendet, passen Sie die Befehle entsprechend an.

Um die von Amazon EC2 SpotFleetTaggingRole verwaltete Richtlinie mit Ihrer Spot-Flottenrolle zu verknüpfen

1. Um die Amazon EC2 SpotFleetTaggingRole Managed IAM-Richtlinie an Ihre *`AmazonEC2SpotFleetRole`* Rolle anzuhängen, führen Sie den folgenden Befehl mit dem AWS CLI aus.

```
$ aws iam attach-role-policy \
```

```
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole
 \
    --role-name AmazonEC2SpotFleetRole
```

2. Um die Amazon EC2 SpotFleetRole Managed IAM-Richtlinie von Ihrer *AmazonEC2SpotFleetRole* Rolle zu trennen, führen Sie den folgenden Befehl mit dem aus. AWS CLI

```
$ aws iam detach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \
    --role-name AmazonEC2SpotFleetRole
```

Secrets Manager können nicht abgerufen werden

Wenn Sie ein AMI mit einem Amazon ECS-Agenten verwenden, der älter als Version 1.16.0-1 ist, müssen Sie die Amazon ECS-Agenten-Konfigurationsvariable verwenden, um diese Funktion nutzen `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` zu können. Sie können es der `./etc/ecs/ecs.config` Datei einer neuen Container-Instance hinzufügen, wenn Sie diese Instance erstellen. Oder Sie können es zu einer vorhandenen Instanz hinzufügen. Wenn Sie es zu einer vorhandenen Instanz hinzufügen, müssen Sie den ECS-Agenten neu starten, nachdem Sie ihn hinzugefügt haben. Weitere Informationen finden Sie unter Amazon ECS Container Agent Configuration im Entwicklerhandbuch zum Amazon Elastic Container Service.

Die Ressourcenanforderungen für die Jobdefinition können nicht außer Kraft gesetzt werden

Die Speicher- und vCPU-Überschreibungen, die in der Struktur `memory` und `vcpus` den Elementen der ContainerOverrides-Struktur angegeben sind, die an übergeben wurden, können die Speicher- und vCPU-Anforderungen SubmitJob, die in der ResourceRequirements-Struktur in der Auftragsdefinition angegeben sind, nicht überschreiben.

Wenn Sie versuchen, diese Ressourcenanforderungen zu überschreiben, wird möglicherweise die folgende Fehlermeldung angezeigt:

„Dieser Wert wurde in einem veralteten Schlüssel übermittelt und kann zu einem Konflikt mit dem Wert führen, der in den Ressourcenanforderungen der Jobdefinition angegeben ist.“

Um dies zu korrigieren, geben Sie die Speicher- und vCPU-Anforderungen im Mitglied ResourceRequirements der ContainerOverrides an. Zum Beispiel, wenn Ihre Speicher- und vCPU-Überschreibungen in den folgenden Zeilen angegeben sind.

```
"containerOverrides": {
   "memory": 8192,
   "vcpus": 4
}
```

Ändern Sie sie wie folgt:

```
"containerOverrides": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "8192"
      },
      {
         "type": "VCPU",
         "value": "4"
      }
   ],
}
```

Nehmen Sie dieselbe Änderung an den Speicher- und vCPU-Anforderungen vor, die im ContainerProperties-Objekt in der Auftragsdefinition angegeben sind. Zum Beispiel, wenn Ihre Speicher- und vCPU-Anforderungen in den folgenden Zeilen angegeben sind.

```
{
   "containerProperties": {
      "memory": 4096,
      "vcpus": 2,
}
```

Ändern Sie sie wie folgt:

```
"containerProperties": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "4096"
```

```
        },
        {
            "type": "VCPU",
            "value": "2"
        }
    ],
}
```

Fehlermeldung beim Aktualisieren der **`desiredvCpus`** Einstellung

Wenn Sie die AWS Batch API verwenden, um die gewünschte v CPUs (`desiredvCpus`) -Einstellung zu aktualisieren, wird die folgende Fehlermeldung angezeigt.

`Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to minvCpus`.

Dieses Problem tritt auf, wenn der aktualisierte `desiredvCpus` Wert unter dem aktuellen `desiredvCpus` Wert liegt. Wenn Sie den `desiredvCpus` Wert aktualisieren, müssen beide der folgenden Bedingungen zutreffen:

- Der `desiredvCpus` Wert muss zwischen den `maxvCpus` Werten `minvCpus` und liegen.
- Der aktualisierte `desiredvCpus` Wert muss größer oder gleich dem aktuellen `desiredvCpus` Wert sein.

AWS Batch auf Amazon EKS

Themen

- INVALIDRechenumgebung
- AWS Batch bei Amazon EKS bleibt der Job im RUNNABLE Status hängen
- AWS Batch bei Amazon EKS bleibt der Job im STARTING Status hängen
- Stellen Sie sicher, dass der richtig konfiguriert aws-auth ConfigMap ist
- RBAC-Berechtigungen oder -Bindungen sind nicht richtig konfiguriert

In den folgenden Themen finden Sie Überprüfungsprozesse und mögliche Lösungen für häufig auftretende Probleme, die bei der Verwendung AWS Batch auf Amazon Elastic Kubernetes Service auftreten können.

INVALIDRechenumgebung

Es ist möglich, dass Sie eine verwaltete Rechenumgebung falsch konfiguriert haben. Wenn Sie das getan haben, wechselt die Computerumgebung in einen `INVALID` Status und kann keine Stellen zur Vermittlung annehmen. In den folgenden Abschnitten werden die möglichen Ursachen beschrieben und wie Sie anhand der Ursache Fehler beheben können.

Nicht unterstützt Kubernetes version

Möglicherweise wird eine Fehlermeldung angezeigt, die der folgenden ähnelt, wenn Sie den `CreateComputeEnvironment` API-Vorgang oder `UpdateComputeEnvironment` den API-Vorgang verwenden, um eine Rechenumgebung zu erstellen oder zu aktualisieren. Dieses Problem tritt auf, wenn Sie eine nicht unterstützte angeben Kubernetes Version in`EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Um dieses Problem zu beheben, löschen Sie die Rechenumgebung und erstellen Sie sie anschließend mit einer unterstützten Kubernetes Version.

Sie können ein kleines Versions-Upgrade auf Ihrem Amazon EKS-Cluster durchführen. Sie können den Cluster beispielsweise von `1.xx` auf aktualisieren, `1.yy` auch wenn die Nebenversion nicht unterstützt wird.

Der Status der Rechenumgebung kann sich jedoch `INVALID` nach einem Update der Hauptversion auf ändern. Dies ist beispielsweise der Fall, wenn Sie ein Upgrade einer Hauptversion von `1.xx` auf durchführen`2.yy`. Wenn die Hauptversion von nicht unterstützt wird AWS Batch, wird eine Fehlermeldung angezeigt, die der folgenden ähnelt.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Um dieses Problem zu beheben, geben Sie ein unterstütztes Kubernetes Version, wenn Sie eine API-Operation verwenden, um eine Rechenumgebung zu erstellen oder zu aktualisieren.

AWS Batch auf Amazon unterstützt EKS derzeit Folgendes Kubernetes Versionen:

- `1.32`
- `1.31`
- `1.30`

- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`

Instanzprofil existiert nicht

Wenn das angegebene Instance-Profil nicht existiert, wird der Status der Datenverarbeitungsumgebung AWS Batch auf Amazon EKS auf geändert`INVALID`. Im `statusReason` Parameter wird ein Fehler angezeigt, der dem folgenden ähnelt.

```
CLIENT_ERROR - Instance profile arn:aws:iam::...:instance-profile/<name> does not exist
```

Um dieses Problem zu beheben, geben Sie ein funktionierendes Instanzprofil an oder erstellen Sie es. Weitere Informationen finden Sie unter IAM-Rolle für Amazon EKS-Knoten im Amazon EKS-Benutzerhandbuch.

Ungültig Kubernetes Namespace

Wenn EKS AWS Batch auf Amazon den Namespace für die Rechenumgebung nicht validieren kann, wird der Status der Rechenumgebung auf `INVALID` geändert. Dieses Problem kann beispielsweise auftreten, wenn der Namespace nicht existiert.

Im `statusReason` Parameter wird eine Fehlermeldung angezeigt, die der folgenden ähnelt.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Dieses Problem kann auftreten, wenn eine der folgenden Bedingungen zutrifft:

- Das Tool Kubernetes Die Namespace-Zeichenfolge im `CreateComputeEnvironment` Aufruf ist nicht vorhanden. Weitere Informationen finden Sie unter CreateComputeEnvironment.
- Die erforderlichen RBAC-Berechtigungen (Role-Based Access Control) zur Verwaltung des Namespaces sind nicht richtig konfiguriert.
- AWS Batch hat keinen Zugriff auf Amazon EKS Kubernetes API-Server-Endpunkt.

Informationen zum Beheben dieses Problems finden Sie unter Stellen Sie sicher, dass der richtig konfiguriert `aws-auth ConfigMap` ist. Weitere Informationen finden Sie unter Erste Schritte mit AWS Batch Amazon EKS.

Computerumgebung gelöscht

Angenommen, Sie löschen einen Amazon EKS-Cluster, bevor Sie die angehängte AWS Batch Amazon EKS-Rechenumgebung löschen. Dann wird der Status der Rechenumgebung auf geändert`INVALID`. In diesem Szenario funktioniert die Rechenumgebung nicht richtig, wenn Sie den Amazon EKS-Cluster mit demselben Namen neu erstellen.

Um dieses Problem zu beheben, löschen Sie die Datenverarbeitungsumgebung AWS Batch auf Amazon EKS und erstellen Sie sie anschließend neu.

Knoten treten dem Amazon EKS-Cluster nicht bei

AWS Batch auf Amazon EKS skaliert eine Rechenumgebung herunter, wenn festgestellt wird, dass nicht alle Knoten dem Amazon EKS-Cluster beigetreten sind. Wenn AWS Batch Amazon EKS die Rechenumgebung herunterskaliert, wird der Status der Rechenumgebung auf geändert`INVALID`.

> **Note**
>
> AWS Batch ändert den Status der Rechenumgebung nicht sofort, sodass Sie das Problem debuggen können.

Im `statusReason` Parameter wird eine Fehlermeldung angezeigt, die einer der folgenden ähnelt:

```
Your compute environment has been INVALIDATED and scaled down because
none of the instances joined the underlying ECS Cluster. Common issues
preventing instances joining are the following: VPC/Subnet configuration
preventing communication to ECS, incorrect Instance Profile policy
preventing authorization to ECS, or customized AMI or LaunchTemplate
configurations affecting ECS agent.
```

```
Your compute environment has been INVALIDATED and scaled down because
none of the nodes joined the underlying Amazon EKS Cluster. Common issues
preventing nodes joining are the following: networking configuration
```

```
preventing communication to Amazon EKS Cluster, incorrect Amazon EKS
Instance Profile or Kubernetes RBAC policy preventing authorization
to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations
affecting Amazon EKS/Kubernetes node bootstrap.
```

Wenn Sie ein standardmäßiges Amazon EKS-AMI verwenden, sind die häufigsten Ursachen für dieses Problem die folgenden:

- Die Instance-Rolle ist nicht richtig konfiguriert. Weitere Informationen finden Sie unter IAM-Rolle für Amazon EKS-Knoten im Amazon EKS-Benutzerhandbuch.
- Die Subnetze sind nicht richtig konfiguriert. Weitere Informationen finden Sie unter Amazon EKS-VPC- und Subnetzanforderungen und Überlegungen im Amazon EKS-Benutzerhandbuch.
- Die Sicherheitsgruppe ist nicht richtig konfiguriert. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zu Amazon EKS-Sicherheitsgruppen im Amazon EKS-Benutzerhandbuch.

> **Note**
>
> Möglicherweise wird Ihnen auch eine Fehlermeldung im Personal Health Dashboard (PHD) angezeigt.

AWS Batch bei Amazon EKS bleibt der Job im **RUNNABLE** Status hängen

Ein `aws-auth ConfigMap` wird automatisch erstellt und auf Ihren Cluster angewendet, wenn Sie eine verwaltete Knotengruppe oder eine Knotengruppe mithilfe von`eksctl`. Eine `aws-auth ConfigMap` wird zunächst erstellt, damit Knoten Ihrem Cluster beitreten können. Sie verwenden den jedoch auch, `aws-auth ConfigMap` um Benutzern und Rollen den Zugriff auf rollenbasierte Zugriffssteuerung (RBAC) hinzuzufügen.

Gehen Sie wie folgt vor, um zu überprüfen, ob der richtig konfiguriert ist `aws-authConfigMap`:

1. Rufen Sie die zugewiesenen Rollen ab in: `aws-auth ConfigMap`

   ```
   $ kubectl get configmap -n kube-system aws-auth -o yaml
   ```

2. Stellen Sie sicher, dass der wie folgt konfiguriert `roleARN` ist.

   ```
   rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
   ```

> **Note**
>
> Sie können auch die Protokolle der Amazon EKS-Kontrollebene überprüfen. Weitere Informationen finden Sie unter Protokollierung der Amazon EKS-Kontrollebene im Amazon EKS-Benutzerhandbuch.

Um ein Problem zu lösen, bei dem ein Job in einem `RUNNABLE` Status hängengeblieben ist, empfehlen wir Ihnen, das Manifest erneut anzuwenden. `kubectl` Weitere Informationen finden Sie unter Schritt 2: Bereiten Sie Ihren Amazon EKS-Cluster vor für AWS Batch. Oder Sie können es verwenden, `kubectl` um das manuell zu bearbeiten. `aws-auth` `ConfigMap` Weitere Informationen finden Sie unter Aktivieren des IAM-Benutzer- und Rollenzugriffs auf Ihren Cluster im Amazon EKS-Benutzerhandbuch.

AWS Batch bei Amazon EKS bleibt der Job im **`STARTING`** Status hängen

Ein Job kann so lange im `STARTING` Status bleiben, wenn der Pod `PENDING` bei `ContainerCreating` lang andauernden Anfragen vom Kubelet (`pull`,, `undattach`) hängen bleibt `logexec`, bis das Pod-Startproblem behoben oder der Job beendet ist. In den unten aufgeführten Qualifizierungsszenarien AWS Batch wird der Job in Ihrem Namen beendet, andernfalls muss der Job manuell über die TerminateJob API beendet werden.

Um zu überprüfen, warum ein Job möglicherweise hängen bleibt`STARTING`, verwenden Sie, Tutorial: Ordnen Sie einen laufenden Job einem Pod und einem Knoten zu um den `podName` Pod zu finden und zu beschreiben:

```
% kubectl describe pod aws-batch.000c8190-87df-31e7-8819-176fe017a24a -n my-aws-batch-
namespace
Name:             aws-batch.000c8190-87df-31e7-8819-176fe017a24a
Namespace:        my-aws-batch-namespace
...
Containers:
  default:
...
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
...
Conditions:
```

```
  Type                        Status
  PodReadyToStartContainers   False
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
...
Events:
  Type     Reason       Age    From     Message
  ----     ------       ----   ----     -------
  Warning  FailedMount  2m32s  kubelet  Unable to attach or mount volumes: ...
```

Erwägen Sie, Ihren EKS-Cluster so zu konfigurieren, dass die Protokolle der Kontrollebene an Logs gesendet CloudWatch werden, um vollständige Transparenz zu gewährleisten.

Szenario: Dauerhafter Fehler beim Anhängen oder Bereitstellen von Volumenansprüchen

Jobs, die Persistent Volume Claims verwenden und bei denen das Volume nicht angehängt oder bereitgestellt werden kann, können beendet werden. Dies kann auf eine falsch konfigurierte Jobdefinition zurückzuführen sein. Weitere Details finden Sie unter Tutorial: Erstellen Sie eine Auftragsdefinition mit einem Knoten auf Amazon EKS-Ressourcen.

Stellen Sie sicher, dass der richtig konfiguriert **`aws-auth ConfigMap`** ist

Gehen Sie wie folgt vor, um zu überprüfen, ob der richtig konfiguriert `aws-auth ConfigMap` ist:

1. Rufen Sie die zugewiesenen Rollen in der `aws-auth ConfigMap` ab.

   ```
   $ kubectl get configmap -n kube-system aws-auth -o yaml
   ```

2. Stellen Sie sicher, dass der wie folgt konfiguriert `roleARN` ist.

   ```
   rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
   ```

   > **Note**
   >
   > Der Pfad `aws-service-role/batch.amazonaws.com/` wurde aus dem ARN der serviceverknüpften Rolle entfernt. Dies liegt an einem Problem mit der `aws-auth`

Konfigurationsübersicht. Weitere Informationen finden Sie unter Rollen mit Pfaden funktionieren nicht, wenn der Pfad in ihrem ARN enthalten ist, in der aws-authconfigmap.

> **Note**
>
> Sie können auch die Protokolle der Amazon EKS-Kontrollebene überprüfen. Weitere Informationen finden Sie unter Protokollierung der Amazon EKS-Kontrollebene im Amazon EKS-Benutzerhandbuch.

Um ein Problem zu lösen, bei dem ein Job in einem `RUNNABLE` Status hängengeblieben ist, empfehlen wir Ihnen, das Manifest erneut anzuwenden. `kubectl` Weitere Informationen finden Sie unter Schritt 2: Bereiten Sie Ihren Amazon EKS-Cluster vor für AWS Batch. Oder Sie können es verwenden, `kubectl` um das manuell zu bearbeiten. `aws-auth` `ConfigMap` Weitere Informationen finden Sie unter Aktivieren des IAM-Benutzer- und Rollenzugriffs auf Ihren Cluster im Amazon EKS-Benutzerhandbuch.

RBAC-Berechtigungen oder -Bindungen sind nicht richtig konfiguriert

Wenn Sie Probleme mit RBAC-Berechtigungen oder -Bindungen haben, überprüfen Sie, ob `aws-batch` Kubernetes Die Rolle kann auf die zugreifen Kubernetes Namespace:

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

Sie können den **`kubectl describe`** Befehl auch verwenden, um die Autorisierungen für eine Clusterrolle anzuzeigen oder Kubernetes Namespace.

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

Es folgt eine Beispielausgabe.

```
Name:         aws-batch-cluster-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
```

```
  Resources                                       Non-Resource URLs  Resource Names
 Verbs
  ---------                                       -----------------  --------------
 -----
  configmaps                                      []                 []
 [get list watch]
  nodes                                           []                 []
 [get list watch]
  pods                                            []                 []
 [get list watch]
  daemonsets.apps                                 []                 []
 [get list watch]
  deployments.apps                                []                 []
 [get list watch]
  replicasets.apps                                []                 []
 [get list watch]
  statefulsets.apps                               []                 []
 [get list watch]
  clusterrolebindings.rbac.authorization.k8s.io  []                 []
 [get list]
  clusterroles.rbac.authorization.k8s.io         []                 []
 [get list]
  namespaces                                      []                 []
 [get]
  events                                          []                 []
 [list]
```

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

Es folgt eine Beispielausgabe.

```
Name:         aws-batch-compute-environment-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources                                 Non-Resource URLs  Resource Names  Verbs
  ---------                                 -----------------  --------------  -----
  pods                                      []                 []              [create
 get list watch delete patch]
  serviceaccounts                           []                 []              [get list]
  rolebindings.rbac.authorization.k8s.io  []                 []              [get list]
  roles.rbac.authorization.k8s.io         []                 []              [get list]
```

Wenden Sie die RBAC-Berechtigungen und -Befehle erneut an, um dieses Problem zu beheben. `rolebinding` Weitere Informationen finden Sie unter Schritt 2: Bereiten Sie Ihren Amazon EKS-Cluster vor für AWS Batch.

Ressource: AWS Batch Servicekontingenten

Die folgende Tabelle enthält die Servicekontingenten AWS Batch , die nicht geändert werden können. Jedes Kontingent ist regionsspezifisch.

Ressource	Kontingent
Maximale Anzahl von Auftragswarteschlangen. Weitere Informationen finden Sie unter Warteschlangen für Job.	50
Maximale Anzahl von Rechenumgebungen in Amazon ECS und Amazon EKS. Weitere Informationen finden Sie unter Rechenumgebungen für AWS Batch.	50
Maximale Anzahl von Rechenumgebungen pro Amazon EKS-Cluster.	5
Maximale Anzahl von Rechenumgebungen für jede Job-Warteschlange	3
Maximale Anzahl von Aufgabenabhängigkeiten für eine Aufgabe	20
Maximale Größe der Auftragsdefinition (für `RegisterJobDefinition` API-Operationen)	24 KiB
Maximale Größe der Auftragsnutzlast (für `SubmitJob` API-Operationen)	30 KiB
Maximale Array-Größe für Array-Aufträge	10000
Maximale Anzahl von Aufträgen im Status `SUBMITTED`	1000000
Maximale Anzahl von Transaktionen pro Sekunde (TPS) für jedes Betriebskonto `SubmitJob`	50
Maximale Anzahl verbrauchbarer Ressourcen	50.000

Je nachdem, wie Sie es verwenden AWS Batch, können zusätzliche Kontingente gelten. Weitere Informationen zu EC2 Amazon-Kontingenten finden Sie unter Amazon EC2 Service Quotas in der Allgemeine AWS-Referenz. Weitere Informationen zu Amazon ECS-Kontingenten finden Sie unter

Amazon ECS Service Quotas in der Allgemeine AWS-Referenz. Weitere Informationen zu Amazon EKS-Kontingenten finden Sie unter Amazon EKS Service Quotas in der Allgemeine AWS-Referenz.

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation seit der ersten Version von AWS Batch beschrieben. Wir aktualisieren die Dokumentation regelmäßig, um das Feedback, das Sie uns senden, einzuarbeiten.

Änderung	Beschreibung	Datum
Fügt Unterstützung für EKS AL 2023 hinzu AMIs	Wie führe ich ein Upgrade von EKS AL2 auf EKS AL2 023 durch	24. Juni 2025
Fügt Unterstützung für einen FireLens ECS Exec-Befehl hinzu	Fügt Unterstützung für einen FireLens ECS Exec-Befehl hinzu.	15. April 2025
Fügt Unterstützung für ressourcenbewusstes Planen hinzu für AWS Batch	Integriert die Unterstützung für ressourcenbewusstes Scheduling AWS Batch für Amazon Elastic Container Service, Amazon Elastic Kubernetes Service und. AWS Fargate	27. Februar 2025
Aktualisierte AWS Batch unterstützte Amazon EKS-Versionen	Die Amazon EKS-Versionen, die das Entfernen von Version 1.22 AWS Batch unterstützen, wurden aktualisiert.	11. März 2024
Aktualisierte AWS Batch unterstützte Amazon EKS-Versionen	Die AWS Batch unterstüt zten Amazon EKS-Versionen wurden auf Version 1.29 aktualisiert.	29. Februar 2024
Automatisierte Auftragsw iederholungen	Das Codebeispiel wurde korrigiert.	29. Februar 2024

Fügt Unterstützung für Multi-Container-Jobs hinzu für AWS Batch	Integriert die Unterstützung für Multi-Container-Jobs AWS Batch für Amazon Elastic Container Service, Amazon Elastic Kubernetes Service und. AWS Fargate	28. Februar 2024
Aktualisierte AWS Batch unterstützte Amazon EKS-Versionen	Die AWS Batch unterstützten Amazon EKS-Versionen wurden auf Version 1.28 aktualisiert	27. Januar 2024
Aktualisiert BatchServiceRolePolicy und AWSBatchServiceRole	BatchServiceRolePolicy Aktualisiert, um Unterstützung für die Beschreibung des Anforderungsverlaufs und der Amazon EC2 Auto Scaling Aktivitäten von Spot Fleet hinzuzufügen. AWSBatchServiceRole Es wurde aktualisiert, um eine Erklärung hinzuzufügen IDs, AWS Batch Berechtigungen zu gewähren `ec2:DescribeSpotFleetRequestHistory` und `autoscaling:DescribeScalingActivities` .	05. Dezember 2023

AWS Batch auf Amazon EKS	AWS Batch fügt Unterstützung für die Ausführung von Jobs auf Amazon EKS-Clustern hinzu.	25. Oktober 2022
Dienstübergreifender verwirrter Deputy Prävention für AWS Batch	AWS Batch bietet jetzt einen Workaround für das Sicherheitsproblem mit verwirrten Stellvertretern, das auftritt, wenn eine Entität (ein Dienst oder ein Konto) von einer anderen Entität gezwungen wird, eine Aktion auszuführen.	6. Juni 2022
Schnittstelle VPC-Endpunkte ()AWS PrivateLink	Unterstützung für die Konfiguration von VPC-Endpunkten mit Schnittstelle hinzugefügt. AWS PrivateLink Das bedeutet, dass Sie eine private Verbindung zwischen Ihrer VPC herstellen können, AWS Batch ohne dass der Zugriff über eine NAT-Instance, eine VPN-Verbindung oder AWS Direct Connect erforderlich ist.	15. April 2022
Verbesserte Updates für die Rechenumgebung	AWS Batch erweiterte Support-Updates für Computerumgebungen.	14. April 2022
AWS verwaltete Richtlinienaktualisierungen — Aktualisierung vorhandener Richtlinien	AWS Batch hat bestehende verwaltete Richtlinien aktualisiert.	6. Dezember 2021

Fair-Share-Terminplanung	AWS Batch fügt Unterstützung für das Hinzufügen von Planungsrichtlinien zu Jobwarteschlangen hinzu.	9. November 2021
Amazon EFS	AWS Batch fügt Unterstützung für das Hinzufügen von Amazon EFS-Dateisystemen zu Ihren Jobdefinitionen hinzu.	1. April 2021
Eine serviceverknüpfte Rolle wurde hinzugefügt	AWS Batch fügt die AWSServiceRoleForBatchdienstbezogene Rolle hinzu.	10. März 2021
AWS Fargate Unterstützung	AWS Batch fügt Unterstützung für die Ausführung von Jobs auf Fargate-Ressourcen hinzu.	3. Dezember 2020
Ressourcen-Markierung	AWS Batch fügt Unterstützung für das Hinzufügen von Metadaten-Tags zu Ihren Rechenumgebungen, Jobdefinitionen, Jobwarteschlangen und Jobs hinzu.	7. Oktober 2020
Secrets	AWS Batch fügt Unterstützung für die Weitergabe von Geheimnissen an Jobs hinzu.	1. Oktober 2020
Protokollierung	AWS Batch fügt Unterstützung für die Angabe zusätzlicher Protokolltreiber für Jobs hinzu.	1. Oktober 2020
Zuweisungsstrategien	AWS Batch fügt Unterstützung für mehrere Strategien zur Auswahl von Instance-Typen hinzu.	16. Oktober 2019

EFA-Unterstützung	AWS Batch fügt Unterstützung für Elastic Fabric Adapter (EFA) -Geräte hinzu.	2. August 2019
GPU-Planung	AWS Batch fügt GPU-Planung hinzu. Mit dieser Funktion können Sie die Anzahl der GPUs einzelnen Aufgaben angeben und die Instanzen entsprechend AWS Batch skalieren.	4. April 2019
parallel Jobs mit mehreren Knoten	AWS Batch fügt Unterstützung für parallel Jobs mit mehreren Knoten hinzu. Mit dieser Funktion können Sie einzelne Jobs ausführen, die sich über mehrere EC2 Amazon-Instances erstrecken.	19. November 2018
Berechtigungen auf Ressourcenebene	AWS Batch unterstützt Berechtigungen auf Ressourcenebene für mehrere API-Operationen.	12. November 2018
Unterstützung für Amazon EC2 Launch-Vorlagen	AWS Batch fügt Unterstützung für die Verwendung von Startvorlagen in Computerumgebungen hinzu.	12. November 2018

AWS Batch Timeouts bei Jobs	AWS Batch fügt Unterstützung für Job-Timeout hinzu. Mit dieser Unterstützung können Sie eine bestimmte Timeout-Dauer für Ihre Jobs konfiguri eren, sodass der Job AWS Batch beendet wird, wenn ein Job länger läuft als er sollte.	5. April 2018
AWS Batch Jobs als Ziele EventBridge	AWS Batch Arbeitsplätze werden als EventBridge Ziele zur Verfügung gestellt. Durch die Erstellung einfacher Regeln können Sie Ereigniss e zuordnen und als Reaktion darauf AWS Batch Jobs einreichen.	1. März 2018
CloudTrail Prüfung für AWS Batch	CloudTrail kann Aufrufe von AWS Batch API-Aktionen prüfen.	10. Januar 2018
Array-Jobs	AWS Batch fügt Unterstüt zung für Array-Jobs hinzu. Sie können Array-Jobs für Parameter-Sweep- und Monte-Carlo-Workloads verwenden.	28. November 2017
Erweitertes Tagging AWS Batch	AWS Batch erweitert die Unterstützung für die Tagging-Funktion. Sie können diese Funktion verwenden, um Tags für Amazon EC2 Spot-Instances anzugeben, die in verwalteten Computeru mgebungen gestartet wurden.	26. Oktober 2017

AWS Batch Event-Stream für EventBridge	AWS Batch fügt den Event-Stream für hinzu EventBridge. Sie können AWS Batch Event Stream verwenden, um nahezu in Echtzeit Benachrichtigungen über den Status von Jobs zu erhalten, die an Ihre Jobwarteschlangen gesendet wurden.	24. Oktober 2017
Automatisierte Auftragswiederholungen	AWS Batch fügt Unterstützung für Auftragswiederholungen hinzu. Mit diesem Update können Sie eine Wiederholungsstrategie auf Ihre Jobs und Jobdefinitionen anwenden, sodass Ihre Jobs automatisch wiederholt werden können, falls sie fehlschlagen.	28. März 2017
AWS Batch allgemeine Verfügbarkeit	AWS Batch wird eingeführt und ist als Mittel konzipiert, mit dem Sie Batch-Computing-Workloads auf dem AWS Cloud ausführen können.	5. Januar 2017

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.