Benutzer-Leitfaden

AWS STCK

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

AWS STCK: Benutzer-Leitfaden

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

Die Handelsmarken und die 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 PCS? ... 1
Konzepte ... 1
Beginnen Sie mit AWS PCS ... 3
Voraussetzungen ... 5
Melde dich an für AWS und erstellen Sie einen Administratorbenutzer ... 5
Installieren Sie das AWS CLI für AWS PCS ... 5
Erforderliche IAM-Berechtigungen ... 6
Benutzen CloudFormation ... 6
Erstellen von VPC und Subnetzen ... 6
Suchen Sie die Standardsicherheitsgruppe für die Cluster-VPC ... 8
Sicherheitsgruppen erstellen ... 8
Erstellen Sie die Sicherheitsgruppen ... 9
Erstellen eines Clusters ... 10
Gemeinsamer Speicher in Amazon EFS erstellen ... 11
Erstellen Sie gemeinsamen Speicher in FSx for Lustre ... 11
Erstellen Sie Compute-Knotengruppen ... 13
Erstellen eines Instance-Profils ... 13
Startvorlagen erstellen ... 15
Erstellen Sie eine Rechenknotengruppe für Anmeldeknoten ... 16
Erstellen Sie eine Rechenknotengruppe für Jobs ... 20
Erstellen einer Warteschlange ... 23
Connect zu Ihrem Cluster her ... 24
Erkunden Sie die Cluster-Umgebung ... 25
Benutzer ändern ... 25
Arbeiten Sie mit gemeinsam genutzten Dateisystemen ... 25
Interagiere mit Slurm ... 26
Führen Sie einen Job mit einem einzelnen Knoten aus ... 27
Führen Sie einen MPI-Job mit mehreren Knoten mit Slurm aus ... 29
Löschen Sie Ihre AWS Ressourcen ... 31
Beginnen Sie mit CloudFormation und AWS PCS ... 35
Verwenden CloudFormation um einen Cluster zu erstellen ... 35
Verbinden mit einem Cluster ... 37
Bereinigen Sie einen Cluster ... 38
Teile einer CloudFormation Vorlage für AWS PCS ... 39

Header ... 40
Metadaten ... 40
Parameters ... 41
Mappings ... 42
Ressourcen ... 43
Outputs ... 49
Vorlagen zum Erstellen eines Beispielclusters ... 50
Cluster ... 53
Einen Cluster erstellen ... 53
Voraussetzungen ... 54
Erstellen Sie ein AWS PCS-Cluster ... 54
Aktualisieren eines Clusters ... 59
Vorteile von Cluster-Updates ... 59
Unterstützte Konfigurationsänderungen ... 59
Einschränkungen ... 60
Voraussetzungen für Cluster-Updates ... 60
Aktualisierungsprozess und Auswirkung auf den Job ... 61
Abrechnung bei Updates ... 61
Aktualisieren eines Clusters ... 61
Aktualisierung der Scheduler-Version ... 64
Häufig gestellte Fragen ... 82
Fehlersuche ... 83
Löschen eines Clusters ... 85
Überlegungen beim Löschen eines AWS PCS-Clusters ... 85
Löschen Sie den Cluster ... 85
Cluster-Größe ... 86
Cluster-Geheimnisse ... 87
Verwenden Sie AWS Secrets Manager um das Cluster-Geheimnis zu finden ... 88
Verwenden Sie AWS PCS, um das Cluster-Geheimnis zu finden ... 89
Hol dir das Geheimnis des Slurm-Clusters ... 90
Kundenseitig verwalteter Schlüssel ... 92
Geheime Rotation ... 94
Knotengruppen berechnen ... 102
Eine Compute-Knotengruppe erstellen ... 103
Voraussetzungen ... 104
Erstellen Sie eine Compute-Knotengruppe in AWS STCK ... 104

Aktualisierung einer Rechenknotengruppe 111
Optionen für die Aktualisierung einer AWS PCS-Rechenknotengruppe 111
Überlegungen beim Aktualisieren einer AWS PCS-Rechenknotengruppe 112
Um eine AWS PCS-Rechenknotengruppe zu aktualisieren 113
Löschen einer Compute-Knotengruppe 116
Überlegungen beim Löschen einer Compute-Knotengruppe 116
Löschen Sie die Compute-Knotengruppe 117
Rufen Sie Details zur Compute-Knotengruppe ab 118
Suchen nach Instanzen der Compute-Knotengruppe 121
Aktionen im Lebenszyklus eines Knotens 123
Lebenszyklusphasen 123
Wie funktioniert die Hinrichtung 125
Konfigurieren Sie Lebenszyklus-Aktionen 126
Beispiele 132
AWS-gepflegte Skripte 135
Bewährte Methoden 143
Fehlerbehebung 144
Skalierung 146
Job-level Skalierung 146
Wie Planung und Skalierung zusammenarbeiten 146
Wie AWS PCS stellt Kapazität für einen Auftrag zur Verfügung 146
Garantiert die Kapazität für große oder zeitkritische Jobs 147
Verwenden von Startvorlagen 149
-Übersicht 149
Erstellen einer grundlegenden Startvorlage 151
Arbeiten mit Amazon EC2-Benutzerdaten 153
Beispiel: Software aus einem Paket-Repository installieren 155
Beispiel: Führen Sie Skripts aus einem S3-Bucket aus 156
Beispiel: Legen Sie globale Umgebungsvariablen fest 157
Beispiel: Führen Sie frühe Startvorgänge aus 158
Beispiel: Verwenden Sie ein EFS-Dateisystem als gemeinsam genutztes Home-Verzeichnis 159
Kapazitätsreservierungen 160
Verwendung ODCRs mit AWS PCS 161
Verwenden I-ODCRs mit AWS STK. 164
Kapazitätsblöcke 167

Nützliche Parameter für Startvorlagen 183
Aktivieren Sie die detaillierte CloudWatch Überwachung 183
Instanz-Metadaten-Service Version 2 (IMDS v2) 184
Warteschlangen 185
Erstellen einer Warteschlange 185
Voraussetzungen 185
Um eine Warteschlange in AWS PCS zu erstellen 186
Eine Warteschlange aktualisieren 188
Überlegungen beim Aktualisieren einer AWS PCS-Warteschlange 188
Um eine AWS PCS-Warteschlange zu aktualisieren 188
Löschen einer Warteschlange 190
Überlegungen beim Löschen einer Warteschlange 191
Lösche die Warteschlange 191
Anmeldeknoten 193
Verwenden einer Compute-Knotengruppe für die Anmeldung 193
Erstellen einer AWS PCS-Rechenknotengruppe für Anmeldeknoten 193
Aktualisierung einer AWS PCS-Compute-Knotengruppe für Login-Knoten 194
Löschen einer AWS PCS-Compute-Knotengruppe für Anmeldeknoten 195
Verwendung eigenständiger Instanzen als Anmeldeknoten 195
Schritt 1 — Rufen Sie die Adresse und das Geheimnis für den AWS Ziel-PCS-Cluster ab ... 196
Schritt 2 — Starten Sie eine EC2-Instanz 197
Schritt 3 — Installieren Sie Slurm auf der Instanz 198
Schritt 4 — Rufen Sie das Cluster-Geheimnis ab und speichern Sie es 198
Schritt 5 — Konfigurieren Sie die Verbindung zum AWS PCS-Cluster 200
Schritt 6 — (Optional) Testen Sie die Verbindung 201
Einen eigenständigen Anmeldeknoten mit mehreren Clustern verbinden 202
Voraussetzungen 204
Skriptcode 205
Verwenden des Skripts 219
Netzwerk 222
VPC- und Subnetz-Anforderungen 222
VPC-Anforderungen und -Überlegungen 222
Subnetz-Anforderungen und -Überlegungen 224
Erstellen einer VPC 225
Voraussetzungen 226
Erstellen Sie eine Amazon VPC 226

Sicherheitsgruppen 231
Anforderungen an Sicherheitsgruppen 232
Mehrere Netzwerkschnittstellen 234
Placement-Gruppen 235
Verwendung des Elastic Fabric Adapter (EFA) 236
Identifizieren Sie EFA-enabled EC2-Instances 237
Erstellen Sie eine Sicherheitsgruppe zur Unterstützung der EFA-Kommunikation 237
(Optional) Erstellen Sie eine Platzierungsgruppe 239
Erstellen oder aktualisieren Sie eine EC2-Startvorlage 239
Erstellen oder aktualisieren Sie Compute-Knotengruppen für EFA 240
(Optional) Testen Sie EFA 240
(Optional) Verwenden Sie eine CloudFormation Vorlage, um eine EFA-enabled Startvorlage zu erstellen 242
Netzwerk-Dateisysteme 245
Überlegungen zur Verwendung von Netzwerkdateisystemen 245
Beispiele für Netzwerk-Mounts 246
Amazon Machine Images (AMIs) 252
PCS-ready DLAMI 252
Was ist enthalten? 252
Finden Sie den aktuellen PCS-ready DLAMI 253
Erhalten Sie Benachrichtigungen für neue Versionen 254
Verwendung mit Infrastructure as Code 255
Auf eine neue Version aktualisieren 256
Verwenden von Beispiel-AMIs 256
Finden Sie aktuelle AWS PCS-Beispiel-AMIs 256
Erfahren Sie mehr über AWS PCS-Beispiel-AMIs 259
Erstellen Sie Ihre eigenen AMIs, die kompatibel sind mit AWS 5 STÜCK 259
Benutzerdefiniert AMIs 259
Schritt 1 — Eine temporäre Instanz starten 260
Schritt 2 — Installieren Sie den AWS PCS-Agent 261
Schritt 3 — Slurm installieren 264
Schritt 4 — (Optional) Zusätzliche Treiber, Bibliotheken und Anwendungssoftware installieren 267
Schritt 5 — Erstellen Sie ein mit AWS PCS kompatibles AMI 268
Schritt 6 — Verwenden Sie das benutzerdefinierte AMI mit einer AWS PCS-Compute-Knotengruppe 268

Schritt 7 — Beenden Sie die temporäre Instanz ... 270
Installationsprogramme zum Erstellen von AMIs ... 271
AWS Installationsprogramm für die PCS-Agent-Software ... 271
Slurm-Installationsprogramm ... 272
Unterstützte Betriebssysteme ... 273
Unterstützte Instance-Typen ... 273
Unterstützte Slurm-Versionen ... 273
Überprüfen Sie die Installationsprogramme anhand einer Prüfsumme ... 273
Versionshinweise für AMIs ... 285
Beispiel-AMIs für x86_64 ... 286
Beispiel-AMIs für Arm64 ... 289
Unterstützte Betriebssysteme ... 294
AWS PCS-Agent-Versionen ... 296
Slurm ... 301
Slurm-Versionen ... 301
Unterstützte Slurm-Versionen in AWS 5 STCK ... 302
Nicht unterstützte Slurm-Versionen in AWS 5 STCK ... 304
Versionshinweise ... 305
Häufig gestellte Fragen ... 310
Slurm-Konfiguration ... 312
Modus ohne Konfiguration ... 312
Benutzerdefinierte Slurm-Einstellungen ... 313
Benutzerdefinierte Cgroup-Einstellungen ... 323
Benutzerdefinierte SlurmDBD-Einstellungen ... 324
Benutzerdefinierte GRES-Einstellungen ... 326
Buchhaltung im Slurm-Bereich ... 330
Buchhaltungseinstellungen ändern ... 331
Die wichtigsten Konzepte ... 331
Rufen Sie die Buchhaltungskonfiguration für eine bestehende ab AWS PCS-Cluster ... 333
Slurm-REST-API ... 334
Häufige Anwendungsfälle ... 334
Anforderungen und Einschränkungen ... 334
REST-API aktivieren ... 335
REST-API-Authentifizierung ... 337
Verwenden Sie die REST-API ... 342
HÄUFIG GESTELLTE FRAGEN ZUR REST-API ... 344

Slurm-Neustart ... 347
Vorteile des Slurm-Neustarts ... 348
Wann sollte der Slurm-Neustart verwendet werden ... 348
Einschränkungen ... 348
Starten Sie einen Rechenknoten neu ... 349
Neustart abbrechen ... 350
Häufig gestellte Fragen ... 351
Fehlerbehebung ... 353
Hardwaretopologie ... 354
CPU-Topologie ... 354
GPU-Finität ... 354
Topologieverhalten nach Slurm-Version ... 355
MIG-Konfiguration ... 358
Voraussetzungen ... 358
Was ein MIG-Knoten bei jedem Start haben muss ... 359
Konfigurieren Sie MIG auf den Instanzen ... 359
Deklarieren Sie die MIG-Profile für Slurm ... 364
Einschränkungen ... 366
SPANK-Plugins ... 367
Installieren Sie SPANK-Plugins ... 367
Konfigurieren Sie SPANK-Plugins ... 368
Häufig gestellte Fragen zu SPANK-Plugins ... 369
Slurm-CLI-Filter-Plugins ... 370
Voraussetzungen ... 370
Einschränkungen und Sicherheitsüberlegungen ... 371
Konfigurieren Sie die CLI-Filter-Plugins ... 371
Verwenden von Amazon S3 zur Bereitstellung eines CLI-Filter-Plugin-Skripts ... 375
Übersetze ein Job Submit-Plugin-Skript ... 377
Häufig gestellte Fragen ... 378
Fehlerbehebung ... 379
Slurm-Metriken ... 381
Voraussetzungen ... 382
Aktivieren Sie den Metrik-Endpunkt ... 382
Verwenden Sie den Metrik-Endpunkt ... 382
Erfassen Sie Metriken mit Managed Prometheus Kollektor ... 383
Voraussetzungen ... 383

Konfigurieren Sie Sicherheitsgruppen für den Collector ... 385
Erstellen Sie den verwalteten Collector ... 388
Finden Sie den Cluster-Controller-Endpunkt ... 390
Vorgeschlagene Scrape-Konfiguration ... 391
Überprüfen Sie die Bereitstellung der Kennzahlen ... 393
Query Slurm Metriken mit PromQL ... 394
Sicherheit ... 397
Datenschutz ... 398
Verschlüsselung im Ruhezustand ... 399
Verschlüsselung während der Übertragung ... 399
Schlüsselverwaltung ... 400
Inter-network Datenschutz im Verkehr ... 400
Verschlüsselung des API-Datenverkehrs ... 401
Verschlüsselung des Datenverkehrs ... 401
KMS-Schlüsselrichtlinie für verschlüsselte EBS-Volumes ... 401
VPC-Schnittstellen-Endpunkte (AWS PrivateLink) ... 408
Überlegungen ... 409
Erstellen eines Schnittstellenendpunkts ... 409
Erstellen einer Endpunktrichtlinie ... 410
Identitäts- und Zugriffsverwaltung ... 411
Zielgruppe ... 411
Authentifizierung mit Identitäten ... 412
Verwalten des Zugriffs mit Richtlinien ... 413
Wie AWS Parallel Computing Service arbeitet mit IAM ... 415
Identity-based Beispiele für politische Maßnahmen ... 421
AWS verwaltete Richtlinien ... 425
Service-linked Rollen ... 428
Rolle EC2 Sport ... 430
Mindestberechtigungen ... 431
Instance-Profile ... 438
Fehlerbehebung ... 442
Compliance-Validierung ... 445
Ausfallsicherheit ... 445
Infrastruktursicherheit ... 446
Schwachstellenanalyse und -management ... 446
Cross-service verwirrte stellvertretende Prävention ... 447

IAM-Rolle für Amazon EC2-Instances, die als Teil einer Rechenknotengruppe bereitgestellt werden ... 449
Bewährte Methoden für die Gewährleistung der Sicherheit ... 450
AMI-related Sicherheit ... 450
Sicherheit im Slurm Workload Manager ... 450
Überwachung und Protokollierung ... 451
Netzwerksicherheit ... 451
Protokollieren und Überwachen ... 452
Protokolle zur Fertigstellung von Aufträgen ... 452
Voraussetzungen ... 453
Richten Sie Protokolle für die Fertigstellung von Aufträgen ein ... 454
Wie finde ich Protokolle zur Fertigstellung von Aufträgen ... 456
Protokollfelder für den Abschluss eines Auftrags ... 456
Beispiele für Protokolle zur Fertigstellung von Aufträgen ... 460
Scheduler-Protokolle ... 465
Voraussetzungen ... 465
Richten Sie Scheduler-Protokolle ein ... 466
Pfade und Namen von Log-Streams im Scheduler ... 468
Beispiel für Scheduler-Protokolldatensätze ... 469
Scheduler-Auditprotokolle ... 470
Voraussetzungen ... 471
Richten Sie Scheduler-Auditprotokolle ein ... 472
Die Pfade und Namen von Log-Streams werden von Scheduler geprüft ... 473
Beispiel für einen Scheduler-Audit-Logeintrag ... 474
Verhalten des Audit-Logs nach Slurm-Version ... 475
Protokolle skalieren ... 476
Voraussetzungen ... 477
Richten Sie Skalierungsprotokolle ein ... 477
Skalierungsprotokolle finden ... 479
Skalieren von Protokollfeldern ... 480
Statuscodes und Ursachencodes ... 481
Beispiel für Skalierungsprotokolle ... 484
Überwachung mit CloudWatch ... 485
Überwachung von Kennzahlen ... 486
Überwachen von Instances ... 486
CloudTrail protokolliert ... 495

AWS PCS-Informationen in CloudTrail ... 496
Grundlegendes zu CloudTrail Protokolldateieinträgen von AWS PCS ... 497
Endpunkte und Servicequoten ... 500
Service-Endpunkte ... 500
Servicekontingente ... 505
Interne Kontingente ... 506
Relevante Kontingente für andere AWS service ... 507
Fehlerbehebung ... 508
Die EC2-Instanz wird nach dem Neustart beendet und ersetzt ... 508
Beheben Sie Probleme mit dem Bootstrap und der Registrierung von Rechenknoten in AWS STCK ... 509
Wie funktioniert Slurm auf AWS 2 STCK ... 510
Rufen Sie Instanzprotokolle ab ... 511
Abrufen VPC/Subnet/Security Gruppen aus einer Instanz-ID ... 512
Probleme bei der Node-Registrierung ... 513
Probleme beim Verbinden des Slurm-Clusters ... 516
Ungültige Knotenregistrierung ... 519
Die CPU-Topologie entspricht nicht der Hardware ... 519
Die GPU-Affinität passt nicht zu den Socket-Grenzen ... 519
Auflösung ... 520
Einreichung des Jobs MaxJobCount Begrenzung ... 520
Fehlende Buchhaltungsunterlagen ... 521
Häufige Ursache ... 521
Auflösung ... 522
Prävention ... 523
Dokumentverlauf ... 524
AWS Glossar ... 577
... dlxxviii

Was ist AWS Parallel Computing Service?

AWS Parallel Computing Service (AWS PCS) ist ein verwalteter Service, der es einfacher macht, HPC-Workloads (High Performance Computing) auszuführen und zu skalieren und wissenschaftliche und technische Modelle für die AWS Verwendung von Slurm zu erstellen. Verwenden Sie AWS PCS, um Rechencluster aufzubauen, die erstklassige AWS Rechen-, Speicher-, Netzwerk- und Visualisierungsfunktionen integrieren. Führen Sie Simulationen durch oder erstellen Sie wissenschaftliche und technische Modelle. Rationalisieren und vereinfachen Sie Ihren Clusterbetrieb mithilfe der integrierten Management- und Observability-Funktionen. Geben Sie Ihren Benutzern die Möglichkeit, sich auf Forschung und Innovation zu konzentrieren, indem Sie ihnen ermöglichen, ihre Anwendungen und Jobs in einer vertrauten Umgebung auszuführen.

Themen

- Konzepte in AWS PCS

Konzepte in AWS PCS

Ein Cluster in AWS PCS hat eine oder mehrere Warteschlangen, die mindestens einer Rechenknotengruppe zugeordnet sind. Jobs werden an Warteschlangen weitergeleitet und auf EC2 Instanzen ausgeführt, die durch Rechenknotengruppen definiert sind. Sie können diese Grundlagen nutzen, um anspruchsvolle HPC-Architekturen zu implementieren.

Cluster

Ein Cluster ist eine Ressource für die Verwaltung von Ressourcen und die Ausführung von Workloads. Ein Cluster ist eine AWS PCS-Ressource, die eine Zusammenstellung von Rechen-, Netzwerk-, Speicher-, Identitäts- und Job-Scheduler-Konfigurationen definiert. Sie erstellen einen Cluster, indem Sie angeben, welchen Job-Scheduler Sie verwenden möchten (derzeit Slurm), welche Scheduler-Konfiguration Sie wünschen, welchen Service Controller Sie für die Verwaltung des Clusters verwenden möchten und in welcher VPC die Cluster-Ressourcen gestartet werden sollen. Der Scheduler akzeptiert und plant Jobs und startet auch die Rechenknoten (EC2 Instanzen), die diese Jobs verarbeiten.

Compute-Knotengruppe

Eine Rechenknotengruppe ist eine Sammlung von Rechenknoten, die AWS PCS verwendet, um Jobs auszuführen oder interaktiven Zugriff auf einen Cluster zu ermöglichen. Wenn Sie eine Compute-

Knotengruppe definieren, geben Sie allgemeine Merkmale wie EC2 Amazon-Instance-Typen, minimale und maximale Instance-Anzahl, Ziel-VPC-Subnetze, Amazon Machine Image (AMI), Kaufoption und benutzerdefinierte Startkonfiguration an. AWS PCS verwendet diese Einstellungen, um Rechenknoten in einer Rechenknotengruppe effizient zu starten, zu verwalten und zu beenden.

Warteschlange

Wenn Sie einen Job auf einem bestimmten Cluster ausführen möchten, senden Sie ihn an eine bestimmte Warteschlange (manchmal auch Partition genannt). Der Job bleibt in der Warteschlange, bis AWS PCS plant, dass er auf einer Rechenknotengruppe ausgeführt wird. Sie ordnen jeder Warteschlange eine oder mehrere Rechenknotengruppen zu. Eine Warteschlange ist erforderlich, um Jobs auf den zugrunde liegenden Compute-Knotengruppenressourcen unter Verwendung verschiedener vom Job-Scheduler angebotener Planungsrichtlinien zu planen und auszuführen. Benutzer reichen Jobs nicht direkt an einen Rechenknoten oder eine Rechenknotengruppe weiter.

Systemadministrator

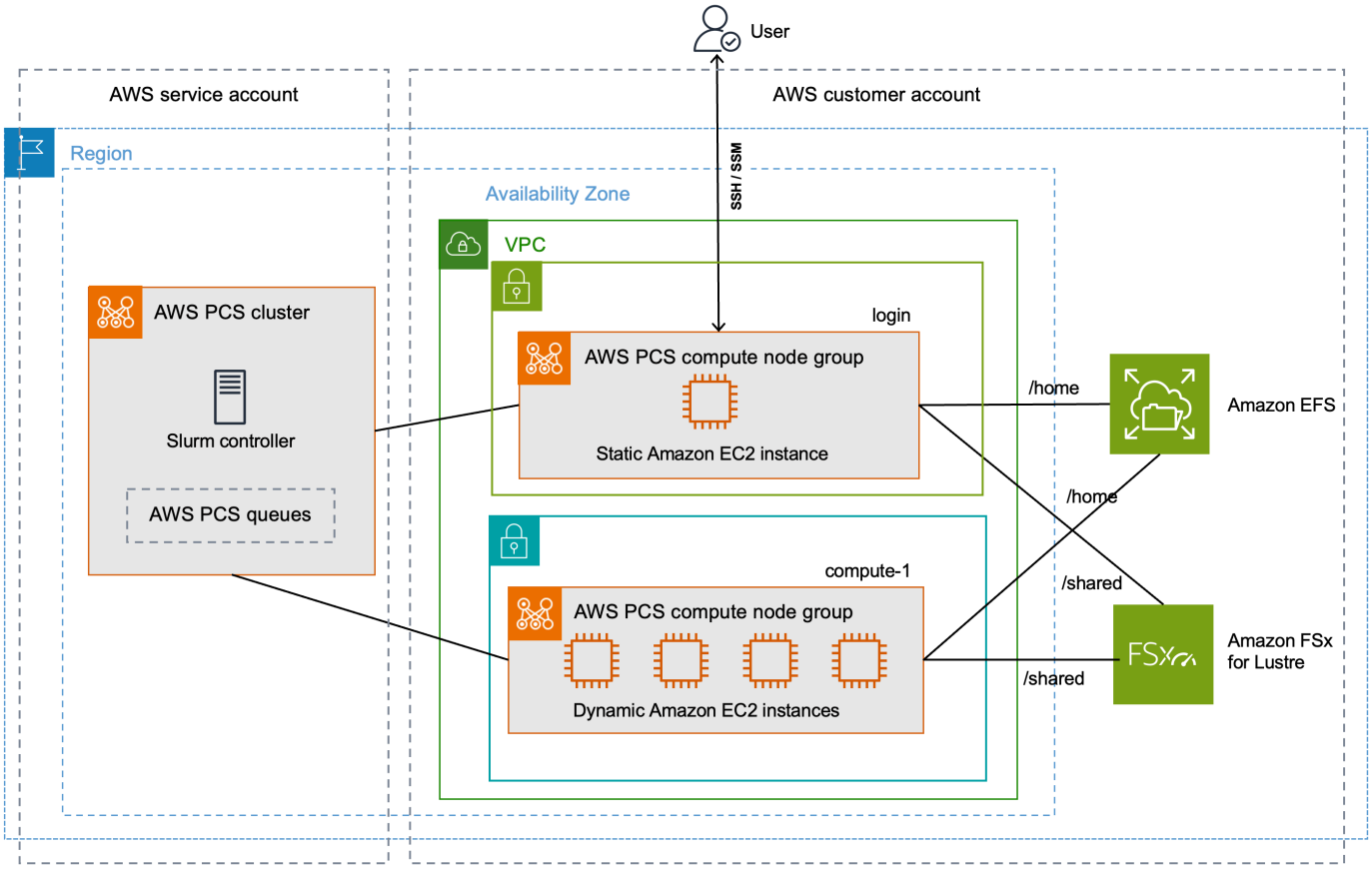
Ein Systemadministrator stellt einen Cluster bereit, verwaltet und betreibt ihn. Sie können über die AWS PCS API und AWS-Managementkonsole das AWS SDK auf AWS PCS zugreifen. Sie haben über SSH Zugriff auf bestimmte Cluster oder können dort administrative Aufgaben ausführen AWS Systems Manager, Jobs ausführen, Daten verwalten und andere Shell-basierte Aktivitäten ausführen. Weitere Informationen finden Sie in der AWS Systems Manager Dokumentation.

Endbenutzer

Ein Endbenutzer ist nicht dafür day-to-day verantwortlich, einen Cluster bereitzustellen oder zu betreiben. Sie verwenden eine Terminalschnittstelle (wie SSH), um auf Clusterressourcen zuzugreifen, Jobs auszuführen, Daten zu verwalten und andere Shell-basierte Aktivitäten durchzuführen.

Fangen Sie an mit AWS Dienst für parallele Datenverarbeitung

Dies ist ein Tutorial zum Erstellen eines einfachen Clusters, mit dem Sie AWS PCS testen können. Die folgende Abbildung zeigt das Design des Clusters.



Das Cluster-Design für das Tutorial besteht aus den folgenden Hauptkomponenten:

- Eine VPC und Subnetze, die die AWS PCS-Netzwerkanforderungen erfüllen.
- Ein Amazon EFS-Dateisystem, das als gemeinsames Home-Verzeichnis verwendet wird.
- Ein Amazon FSx for Lustre-Dateisystem, das ein gemeinsam genutztes Hochleistungsverzeichnis bereitstellt.
- Ein AWS PCS-Cluster, der einen Slurm-Controller bereitstellt.
- 2 AWS PCS-Rechenknotengruppen.
  - Die `login` Knotengruppe, die einen Shell-basierten interaktiven Zugriff auf das System ermöglicht.

  - Die `compute-1` Knotengruppe bietet elastisch skalierbare Instanzen zur Ausführung von Jobs.
- 1 Warteschlange, die Jobs an EC2-Instances in der `compute-1` Knotengruppe sendet.

Der Cluster benötigt zusätzliche AWS Ressourcen, wie Sicherheitsgruppen, IAM-Rollen und EC2-Startvorlagen, die im Diagramm nicht dargestellt sind.

> **Note**
>
> Wir empfehlen, dass Sie die Befehlszeilenschritte in diesem Thema in einer Bash-Shell ausführen. Wenn Sie keine Bash-Shell verwenden, erfordern einige Skriptbefehle wie Zeilenfortsetzungszeichen und die Art und Weise, wie Variablen gesetzt und verwendet werden, eine Anpassung für Ihre Shell. Darüber hinaus können die Zitier- und Escape-Regeln für Ihre Shell unterschiedlich sein. Weitere Informationen finden Sie unter Anführungszeichen und Literale mit Zeichenfolgen AWS CLI im AWS Command Line Interface Benutzerhandbuch für Version 2.

Themen

- Voraussetzungen für den Einstieg in AWS STK.
- Verwendung AWS CloudFormation mit dem AWS PCS-Tutorial
- Erstellen Sie eine VPC und Subnetze für PCS AWS
- Sicherheitsgruppen für AWS PCS erstellen
- Erstellen Sie einen Cluster in AWS STCK
- Erstellen Sie gemeinsam genutzten Speicher für AWS PCS in Amazon Elastic File System
- Erstellen Sie gemeinsamen Speicher für AWS PCS in Amazon FSx für Lustre
- Erstellen Sie Compute-Knotengruppen in AWS PCS
- Erstellen Sie eine Warteschlange zur Verwaltung von Jobs in AWS PCS
- Connect zu Ihrem AWS PCS-Cluster her
- Erkunden Sie die Cluster-Umgebung in AWS PCS
- Führen Sie einen Einzelknotenjob in AWS PCS aus
- Führen Sie einen MPI-Job mit mehreren Knoten mit Slurm in PCS aus AWS
- Lösche deine AWS Ressourcen für AWS STCK

Voraussetzungen für den Einstieg in AWS STK.

Lesen Sie die folgenden Themen, um Ihre AWS-Konto und Ihre lokale Entwicklungsumgebung für AWS PCS vorzubereiten.

Themen

- Melde dich an für AWS und erstellen Sie einen Administratorbenutzer
- Installieren Sie das AWS CLI für AWS PCS
- Erforderliche IAM-Berechtigungen für AWS STK.

Melde dich an für AWS und erstellen Sie einen Administratorbenutzer

Führen Sie die folgenden Aufgaben aus, um Folgendes einzurichten AWS Dienst für parallele Datenverarbeitung (AWS STÜCK).

Topics

- Melde dich an für AWS-Konto

Melde dich an für AWS-Konto

Um loszulegen mit AWS, du brauchst eine AWS-Konto. Es gibt zwei Möglichkeiten, sich anzumelden AWS. Wenn Sie die vollständige Kontrolle über die Kontokonfiguration und den Speicherort Ihrer Inhalte haben möchten oder wenn Sie regulierte Workloads wie HIPAA oder FedRAMP haben, verwenden Sie Melden Sie sich an für AWS (fortgeschritten). Wenn Sie ein neuer Benutzer sind und ein Konto verwenden möchten, das Sie bereits besitzen, um sich anzumelden AWS und erstellen Sie eine AWS Umgebung mit vorkonfigurierten Standardwerten in select AWS-Regionen, verwenden Melde dich an für AWS.

Installieren Sie das AWS CLI für AWS PCS

Sie müssen die neueste Version von verwenden AWS CLI. Weitere Informationen finden Sie unter Installation oder Aktualisierung auf die neueste Version von AWS CLI im AWS Command Line Interface Benutzerhandbuch für Version 2.

Sie müssen das konfigurieren AWS CLI. Weitere Informationen finden Sie unter Configure the AWS CLI im AWS Command Line Interface Benutzerhandbuch für Version 2.

Geben Sie an der Befehlszeile den folgenden Befehl ein, um Ihre AWS CLI Daten zu überprüfen. Es sollten Hilfeinformationen angezeigt werden.

```
aws pcs help
```

Erforderliche IAM-Berechtigungen für AWS STK.

Der von Ihnen verwendete IAM-Sicherheitsprinzipal muss über Berechtigungen für die Arbeit mit AWS PCS-IAM-Rollen, serviceverknüpften Rollen AWS CloudFormation, einer VPC und verwandten Ressourcen verfügen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für AWS Dienst für parallele Datenverarbeitung und Erstellen einer serviceverknüpften Rolle im Benutzerhandbuch.AWS Identity and Access Management Sie müssen alle Schritte in diesem Handbuch als derselbe Benutzer ausführen. Führen Sie den folgenden Befehl aus, um den aktuellen Benutzer zu überprüfen:

```
aws sts get-caller-identity
```

Verwendung AWS CloudFormation mit dem AWS PCS-Tutorial

Das AWS PCS-Tutorial besteht aus vielen Schritten und soll Ihnen helfen, die Bestandteile eines AWS PCS-Clusters und die zu seiner Erstellung erforderlichen Verfahren zu verstehen. Wir empfehlen, dass Sie die Schritte des Tutorials mindestens einmal durchführen. Sobald Sie ein gutes Verständnis dafür haben, AWS CloudFormation worum es geht, können Sie den Beispielcluster mithilfe von Automatisierung schnell erstellen.

CloudFormation ist ein AWS Service, mit dem Sie AWS Infrastrukturbereitstellungen vorhersehbar und wiederholt erstellen und bereitstellen können. Sie können eine CloudFormation Vorlage verwenden, um die AWS Ressourcen für den Beispielcluster automatisch als einzelne Einheit, einen sogenannten Stack, bereitzustellen. Sie können den Stapel löschen, wenn Sie damit fertig sind.

Weitere Informationen finden Sie unter Erste Schritte mit CloudFormation AWS PCS.

Erstellen Sie eine VPC und Subnetze für PCS AWS

Sie können eine VPC und Subnetze mit einer CloudFormation Vorlage erstellen. Verwenden Sie die folgende URL, um die CloudFormation Vorlage herunterzuladen, und laden Sie sie dann in die CloudFormation Konsole hoch, um einen neuen CloudFormation Stack zu erstellen. Weitere

Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Verwenden der CloudFormation Konsole.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

Geben Sie bei geöffneter Vorlage in der CloudFormation Konsole die folgenden Optionen ein. Sie können die in der Vorlage bereitgestellten Standardwerte verwenden.

- Gehen Sie unter Geben Sie einen Stacknamen ein:
  - Geben Sie unter Stackname Folgendes ein:

    ```
    hpc-networking
    ```
- Unter Parameter:
  - Unter VPC:
    - Geben Sie unter CidrBlockFolgendes ein:

      ```
      10.3.0.0/16
      ```
  - Unter Subnetze A:
    - Geben Sie unter CidrPublicSubnetA Folgendes ein:

      ```
      10.3.0.0/20
      ```
    - Geben Sie unter CidrPrivateSubnetA Folgendes ein:

      ```
      10.3.128.0/20
      ```
  - Unter Subnetze B:
    - Geben Sie unter CidrPublicSubnetB Folgendes ein:

      ```
      10.3.16.0/20
      ```
    - Geben Sie unter CidrPrivateSubnetB Folgendes ein:

      ```
      10.3.144.0/20
      ```
  - Unter Subnetze C:

    - Wählen Sie für ProvisionSubnetsC die Option True
    - Geben Sie unter CidrPublicSubnetC Folgendes ein:

      ```
      10.3.32.0/20
      ```

    - Geben Sie unter CidrPrivateSubnetC Folgendes ein:

      ```
      10.3.160.0/20
      ```

- Unter Fähigkeiten:
    - Markieren Sie das Kästchen Ich bestätige, dass dadurch IAM-Ressourcen erstellt werden AWS CloudFormation könnten.

Überwachen Sie den Status des CloudFormation Stacks. Wenn es erreicht ist`CREATE_COMPLETE`, suchen Sie die ID für die Standardsicherheitsgruppe in der neuen VPC. Sie verwenden die ID später im Tutorial.

Suchen Sie die Standardsicherheitsgruppe für die Cluster-VPC

Gehen Sie wie folgt vor, um die ID für die Standardsicherheitsgruppe in der neuen VPC zu finden:

- Navigieren Sie zur Amazon VPC-Konsole.
- Wählen Sie im VPC-Dashboard die Option Nach VPC filtern aus.
    - Wählen Sie die VPC aus, mit `hpc-networking` der der Name beginnt.
    - Wählen Sie unter Sicherheit die Option Sicherheitsgruppen aus.
- Suchen Sie die Sicherheitsgruppen-ID für die angegebene Gruppe`default`. Sie hat die Beschreibung`default VPC security group`. Sie verwenden die ID später, um EC2-Startvorlagen zu konfigurieren.

Sicherheitsgruppen für AWS PCS erstellen

AWS PCS stützt sich auf Sicherheitsgruppen, um den Netzwerkverkehr in und aus einem Cluster und seinen Compute-Knotengruppen zu verwalten. Ausführliche Informationen zu diesem Thema finden Sie unterAnforderungen und Überlegungen zur Sicherheitsgruppe.

In diesem Schritt verwenden Sie eine CloudFormation Vorlage, um zwei Sicherheitsgruppen zu erstellen.

- Eine Cluster-Sicherheitsgruppe, die die Kommunikation zwischen AWS PCS-Controllern, Rechenknoten und Anmeldeknoten ermöglicht.
- Eine SSH-Sicherheitsgruppe für eingehende Nachrichten, die Sie optional zu Ihren Anmeldeknoten hinzufügen können, um den SSH-Zugriff zu unterstützen

Erstellen Sie die Sicherheitsgruppen für PCS AWS

Sie können eine CloudFormation Vorlage verwenden, um die Sicherheitsgruppen zu erstellen. Verwenden Sie die folgende URL, um die CloudFormation Vorlage herunterzuladen, und laden Sie sie dann in die CloudFormation Konsole hoch, um einen neuen CloudFormation Stack zu erstellen. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Verwenden der CloudFormation Konsole.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-cluster-sg.yaml
```

Geben Sie bei geöffneter Vorlage in der AWS CloudFormation Konsole die folgenden Optionen ein. Beachten Sie, dass einige Optionen in der Vorlage bereits ausgefüllt sind. Sie können sie einfach als Standardwerte beibehalten.

- Unter Geben Sie einen Stacknamen an
  - Geben Sie unter Stackname Folgendes ein:

    ```
    getstarted-sg
    ```
- Unter Parameter
  - Wählen Sie VpcIdunter die VPC aus, mit `hpc-networking` der der Name beginnt.
  - (Optional) Geben Sie unter ClientIpCidreinen restriktiveren IP-Bereich für die eingehende SSH-Sicherheitsgruppe ein. Wir empfehlen, dass Sie dies mit Ihrer eigenen IP/Ihrem eigenen Subnetz einschränken (x.x.x.x/32 für Ihre eigene IP oder x.x.x.x/24 für den Bereich). Ersetzen Sie x.x.x.x durch Ihre eigene ÖFFENTLICHE IP. Sie können Ihre öffentliche IP mithilfe von Tools wie https://ifconfig.co/ abrufen.

Überwachen Sie den Status des CloudFormation Stacks. Wenn es `CREATE_COMPLETE` die Sicherheitsgruppe erreicht, sind die Ressourcen bereit.

Es wurden zwei Sicherheitsgruppen mit den folgenden Namen erstellt:

- `cluster-getstarted-sg`— das ist die Cluster-Sicherheitsgruppe
- `inbound-ssh-getstarted-sg`— Dies ist eine Sicherheitsgruppe, die eingehenden SSH-Zugriff ermöglicht

Erstellen Sie einen Cluster in AWS STCK

In AWS PCS ist ein Cluster eine persistente Ressource für die Verwaltung von Ressourcen und die Ausführung von Workloads. Sie erstellen einen Cluster für einen bestimmten Scheduler (AWS PCS unterstützt derzeit Slurm) in einem Subnetz einer neuen oder vorhandenen VPC. Der Cluster akzeptiert und plant Jobs und startet auch die Rechenknoten (EC2-Instances), die diese Jobs verarbeiten.

Um Ihren Cluster zu erstellen

1. Öffnen Sie die AWS PCS-Konsole und wählen Sie Create cluster aus.
2. Geben Sie im Abschnitt Cluster-Details die folgenden Felder ein:
   - Clustername — Geben Sie ein `get-started`
   - Scheduler — Wählen Sie Slurm Version 25.11
   - Controller-Größe — Wählen Sie Klein. Die Controller-Größe bestimmt die Kapazität des verwalteten Slurm-Controllers; klein ist genug für einen Demonstrationscluster.
3. Wählen Sie im Abschnitt Netzwerk Werte für die folgenden Felder aus:
   - VPC — Wählen Sie die benannte VPC `hpc-networking:Large-Scale-HPC`
   - Subnetz — Wählen Sie das Subnetz aus, mit dem der Name beginnt. `hpc-networking:PrivateSubnetA` Der Controller läuft in einem privaten Subnetz, da er nicht über das Internet erreichbar sein muss.
   - Sicherheitsgruppen — Wählen Sie die Cluster-Sicherheitsgruppe mit dem Namen aus `cluster-getstarted-sg`
4. Wählen Sie Cluster erstellen.

> **Note**
>
> Im Feld Status wird während der Bereitstellung des Clusters erstellt. Die Clustererstellung kann mehrere Minuten dauern.

Erstellen Sie gemeinsam genutzten Speicher für AWS PCS in Amazon Elastic File System

Amazon Elastic File System (Amazon EFS) ist ein AWS Service, der serverlosen, vollständig elastischen Dateispeicher bereitstellt, sodass Sie Dateidaten gemeinsam nutzen können, ohne Speicherkapazität und Leistung bereitstellen oder verwalten zu müssen. Weitere Informationen finden Sie unter Was ist Amazon Elastic File System? im Amazon Elastic File System-Benutzerhandbuch.

Der AWS PCS-Demonstrationscluster verwendet ein EFS-Dateisystem, um ein gemeinsames Basisverzeichnis zwischen den Clusterknoten bereitzustellen. Erstellen Sie ein EFS-Dateisystem in derselben VPC wie Ihr Cluster.

So erstellen Sie ein Amazon-EFS-Dateisystem

1. Gehen Sie zur Amazon EFS-Konsole.
2. Stellen Sie sicher, dass sie auf die gleiche Einstellung eingestellt ist AWS-Region , auf der Sie AWS PCS ausprobieren möchten.
3. Wählen Sie Create file system (Dateisystem erstellen) aus.
4. Stellen Sie auf der Seite Dateisystem erstellen die folgenden Parameter ein:
   - Für Name geben Sie `getstarted-efs` ein.
   - Wählen Sie unter Virtual Private Cloud (VPC) die VPC mit dem Namen `hpc-networking:Large-Scale-HPC`
   - Wählen Sie Create (Erstellen) aus. Dadurch kehren Sie zur Seite Dateisysteme zurück.
5. Notieren Sie sich die Dateisystem-ID für das `getstarted-efs` Dateisystem. Sie benötigen diese Informationen später.

Erstellen Sie gemeinsamen Speicher für AWS PCS in Amazon FSx für Lustre

Amazon FSx for Lustre macht es einfach und kostengünstig, das beliebte, leistungsstarke Lustre-Dateisystem zu starten und auszuführen. Sie verwenden Lustre für Workloads, bei denen es auf Geschwindigkeit ankommt, wie maschinelles Lernen, Hochleistungsrechnen (HPC), Videoverarbeitung und Finanzmodellierung. Weitere Informationen finden Sie unter Was ist Amazon FSx for Lustre? im Amazon FSx for Lustre-Benutzerhandbuch.

Der AWS PCS-Demonstrationscluster kann ein FSx for Lustre-Dateisystem verwenden, um ein leistungsstarkes gemeinsames Verzeichnis zwischen den Clusterknoten bereitzustellen. Erstellen Sie ein FSx for Lustre-Dateisystem in derselben VPC wie Ihr Cluster.

Um Ihr FSx for Lustre-Dateisystem zu erstellen

1. Gehen Sie zur Amazon FSx-Konsole.
2. Stellen Sie sicher, dass die Konsole so eingerichtet ist, dass sie dasselbe AWS-Region wie Ihr Cluster verwendet.
3. Wählen Sie Create file system (Dateisystem erstellen) aus.
   - Wählen Sie für Select file system type die Option Amazon FSx for Lustre und dann Next aus.
4. Stellen Sie auf der Seite „Dateisystemdetails angeben“ die folgenden Parameter ein:
   - Unter Dateisystemdetails
     - Für Name geben Sie `getstarted-fsx` ein.
     - Wählen Sie als Bereitstellungs- und Speichertyp die Option Persistent, SSD aus. Persistente Dateisysteme speichern Daten dauerhaft, was für ein gemeinsam genutztes Verzeichnis geeignet ist, das einzelne Rechenknoten überdauert.
     - Wählen Sie für den Durchsatz pro Speichereinheit den Wert 125 MB/s/TiB
     - Geben Sie für Speicherkapazität 1,2 TiB ein
     - Wählen Sie für Metadatenkonfiguration die Option Automatisch
     - Wählen Sie als Datenkomprimierungstyp LZ4
   - Unter Netzwerk und Sicherheit
     - Wählen Sie für Virtual Private Cloud (VPC) die VPC mit dem Namen `hpc-networking:Large-Scale-HPC`
     - Belassen Sie bei VPC-Sicherheitsgruppen den Namen der Sicherheitsgruppe `default`
     - Wählen Sie für Subnetz das Subnetz aus, mit dem der Name beginnt. `hpc-networking:PrivateSubnetA` Platzieren Sie das Dateisystem im selben Subnetz wie Ihre Rechenknoten, sodass sie es über das lokale Netzwerk erreichen.
   - Belassen Sie die anderen Optionen auf ihren Standardwerten.
   - Wählen Sie Weiter aus.
5. Wählen Sie auf der Seite Überprüfen und erstellen die Option Dateisystem erstellen aus. Dadurch kehren Sie zur Seite Dateisysteme zurück.

6. Navigieren Sie zur Detailseite für das FSx for Lustre-Dateisystem, das Sie erstellt haben.
7. Notieren Sie sich den DNS-Namen und den Mount-Namen. Sie geben beide Werte später an, wenn Sie die Node-Lifecycle-Aktion konfigurieren, die dieses Dateisystem mountet. Das `mount-fsx-lustre` Skript mountet das Dateisystem anhand seines DNS-Namens, nicht anhand seiner Dateisystem-ID.

> **Note**
>
> Im Feld Status wird Creating angezeigt, während das Dateisystem bereitgestellt wird. Die Erstellung des Dateisystems kann mehrere Minuten dauern. Warten Sie, bis der Vorgang abgeschlossen ist, bevor Sie mit dem Rest des Tutorials fortfahren.

Erstellen Sie Compute-Knotengruppen in AWS PCS

Eine Rechenknotengruppe ist eine virtuelle Sammlung von Rechenknoten (EC2-Instances), die AWS PCS startet und verwaltet. Wenn Sie eine Compute-Knotengruppe definieren, geben Sie allgemeine Merkmale wie EC2-Instance-Typen, minimale und maximale Instance-Anzahl, Ziel-VPC-Subnetze, bevorzugte Kaufoption und benutzerdefinierte Startkonfiguration an. AWS PCS startet, verwaltet und beendet Rechenknoten in einer Compute-Knotengruppe gemäß diesen Einstellungen auf effiziente Weise. Der Demonstrationscluster verwendet eine Rechenknotengruppe, um Anmeldeknoten für den Benutzerzugriff bereitzustellen, und eine separate Rechenknotengruppe, um Jobs zu verarbeiten. In den folgenden Themen werden die Verfahren zum Einrichten dieser Compute-Knotengruppen in Ihrem Cluster beschrieben.

Themen

- Erstellen Sie ein Instanzprofil für AWS STCK
- Erstellen Sie Startvorlagen für AWS 5 STÜCK
- Erstellen Sie eine Rechenknotengruppe für Anmeldeknoten in AWS STCK
- Erstellen Sie eine Rechenknotengruppe für die Ausführung von Rechenaufträgen in AWS STCK

Erstellen Sie ein Instanzprofil für AWS STCK

Compute-Knotengruppen benötigen ein Instanzprofil, wenn sie erstellt werden. Wenn Sie die AWS-Managementkonsole verwenden, um eine Rolle für Amazon EC2 zu erstellen, erstellt die Konsole

automatisch ein Instanceprofil und gibt ihm denselben Namen wie der Rolle. Weitere Informationen finden Sie im AWS Identity and Access Management Benutzerhandbuch unter Verwenden von Instanzprofilen.

Im folgenden Verfahren verwenden Sie die, AWS-Managementkonsole um eine Rolle für Amazon EC2 zu erstellen, wodurch auch das Instance-Profil für Ihre Compute-Knotengruppen erstellt wird.

Um das Rollen- und Instanzprofil zu erstellen

- Navigieren Sie zur IAM-Konsole.
- Wählen Sie unter Access management (Zugriffsverwaltung) Roles (Rollen) aus.
- Wählen Sie Rolle erstellen aus.
- Unter Vertrauenswürdige Entität auswählen:
  - Wählen Sie als Vertrauenswürdigen Entitätstyp die Option AWS Dienst aus
  - Wählen Sie unter Anwendungsfall die Option EC2 aus.
    - Wählen Sie dann unter Wählen Sie einen Anwendungsfall für den angegebenen Dienst die Option EC2 aus.
  - Wählen Sie Weiter aus.
- Unter Berechtigungen hinzufügen:
  - Suchen Sie in den Berechtigungsrichtlinien nach AWSPCSComputeNodePolicy. Diese AWS verwaltete Richtlinie gewährt die eine Berechtigung, die ein Knoten benötigt, um sich bei Ihrem AWS PCS-Cluster zu registrieren und eine Verbindung zu diesem herzustellen.
  - Markieren Sie das nebenstehende Kästchen AWSPCSComputeNodePolicy, um es der Rolle hinzuzufügen.
  - Suchen Sie in den Berechtigungsrichtlinien nach AmazonSSMManagedInstanceCore. Mit dieser AWS verwalteten Richtlinie können Sie mit Amazon EC2 Systems Manager (SSM) eine Verbindung zu Knoten herstellen, sodass Sie diese erreichen können, ohne SSH öffnen oder Schlüsselpaare verwalten zu müssen.
  - Markieren Sie das Kästchen daneben AmazonSSMManagedInstanceCore, um es der Rolle hinzuzufügen.
  - Suchen Sie in den Berechtigungsrichtlinien nach AmazonS3ReadOnlyAccess. Diese AWS verwaltete Richtlinie ermöglicht es Knoten, die Aktionsskripte für den Node-Lebenszyklus von Amazon S3 herunterzuladen, einschließlich der Skripte, die das AWS Mounten von Speichern und Weiterleiten von Protokollen verwalten.

- Markieren Sie das nebenstehende Kästchen AmazonS3ReadOnlyAccess, um es der Rolle hinzuzufügen.
- Suchen Sie in den Berechtigungsrichtlinien nach CloudWatchAgentServerPolicy. Diese AWS verwaltete Richtlinie ermöglicht es der `configure-cloudwatch-logs` Node-Lifecycle-Aktion, Log-Streams zu erstellen und die Lifecycle-Aktionsprotokolle jedes Knotens an Amazon CloudWatch Logs weiterzuleiten.
- Markieren Sie das nebenstehende Kästchen CloudWatchAgentServerPolicy, um es der Rolle hinzuzufügen.
- Wählen Sie Weiter aus.

- Prüfen und erstellen Sie unter Name:
  - Unter Rollendetails:
    - Geben Sie für Role name (Rollenname) den Namen `AWSPCS-getstarted-role` ein.
  - Wählen Sie Rolle erstellen aus.

Erstellen Sie Startvorlagen für AWS 5 STÜCK

Wenn Sie eine Compute-Knotengruppe erstellen, stellen Sie eine EC2-Startvorlage bereit, die AWS PCS zur Konfiguration der von PCS gestarteten EC2-Instances verwendet. Die Startvorlage definiert Einstellungen wie Sicherheitsgruppen und das SSH-Schlüsselpaar für den Zugriff. Das Mounten des Dateisystems erfolgt separat durch Aktionen im Lebenszyklus von Knoten, die Sie beim Erstellen der Knotengruppe konfigurieren (siehe Erstellen Sie eine Rechenknotengruppe für Anmeldeknoten undErstellen Sie eine Rechenknotengruppe für Jobs).

In diesem Schritt wird eine CloudFormation Vorlage verwendet, um zwei EC2-Startvorlagen zu erstellen. Eine Vorlage wird verwendet, um Anmeldeknoten zu erstellen, und die andere wird verwendet, um Rechenknoten zu erstellen. Der Hauptunterschied zwischen ihnen besteht darin, dass die Anmeldeknoten so konfiguriert werden können, dass sie eingehenden SSH-Zugriff ermöglichen.

Greifen Sie auf die Vorlage zu CloudFormation

Verwenden Sie die folgende URL, um die CloudFormation Vorlage herunterzuladen, und laden Sie die Vorlage dann in die CloudFormation Konsole hoch, um einen neuen CloudFormation Stack zu erstellen. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Verwenden der CloudFormation Konsole.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/getting_started/assets/pcs-lt-efs-fsxl.yaml
```

Verwenden Sie die CloudFormation Vorlage, um EC2-Startvorlagen zu erstellen

Gehen Sie wie folgt vor, um die CloudFormation Vorlage in der CloudFormation Konsole fertigzustellen

- Unter Geben Sie einen Stacknamen ein:
  - Geben Sie unter Stackname den Wert ein`getstarted-lt`.
- Unter Parameter:
  - Unter Sicherheit
    - Wählen Sie für die Sicherheitsgruppe aus VpcSecurityGroupId, die `default` in Ihrer Cluster-VPC benannt ist.
    - Wählen Sie für ClusterSecurityGroupId die Gruppe mit dem Namen `cluster-getstarted-sg`
    - Wählen Sie für SshSecurityGroupId die Gruppe mit dem Namen `inbound-ssh-getstarted-sg`
    - Wählen Sie für SshKeyName Ihr bevorzugtes SSH-Schlüsselpaar aus.
- Wählen Sie Weiter und dann erneut Weiter.
- Wählen Sie Absenden aus.

Überwachen Sie den Status des CloudFormation Stacks. Wenn `CREATE_COMPLETE` die Startvorlage erreicht ist, kann sie verwendet werden.

> **Note**
>
> Um alle Ressourcen zu sehen, die die CloudFormation Vorlage erstellt hat, öffnen Sie die CloudFormation Konsole. Wählen Sie das `getstarted-lt`-Stack, und wählen Sie dann die Registerkarte Ressourcen.

Erstellen Sie eine Rechenknotengruppe für Anmeldeknoten in AWS STCK

Eine Rechenknotengruppe ist eine virtuelle Sammlung von Rechenknoten (EC2-Instances), die AWS PCS startet und verwaltet. Wenn Sie eine Rechenknotengruppe definieren, geben Sie allgemeine

Merkmale wie EC2-Instance-Typen, minimale und maximale Instance-Anzahl, Ziel-VPC-Subnetze, bevorzugte Kaufoption und benutzerdefinierte Startkonfiguration an. AWS PCS startet, verwaltet und beendet Rechenknoten in einer Rechenknotengruppe gemäß diesen Einstellungen effizient.

In diesem Schritt starten Sie eine statische Rechenknotengruppe, die interaktiven Zugriff auf den Cluster bietet. Sie können SSH oder Amazon EC2 Systems Manager (SSM) verwenden, um sich dort anzumelden, dann Shell-Befehle ausführen und Slurm-Jobs verwalten.

Um die Compute-Knotengruppe zu erstellen

- Öffnen Sie die AWS PCS-Konsole und navigieren Sie zu Clusters.
- Wählen Sie den Cluster mit dem Namen `get-started`
- Navigieren Sie zu Compute Node Groups und wählen Sie Create aus.
- Geben Sie im Abschnitt zur Einrichtung der Compute-Knotengruppe Folgendes an:
  - Name der Compute-Knotengruppe — Geben Sie ein`login`.
- Geben Sie unter Computerkonfiguration die folgenden Werte ein oder wählen Sie sie aus:
  - EC2-Startvorlage — Wählen Sie die Startvorlage aus, die den Namen enthält `login-getstarted-lt`
  - IAM-Instanzprofil — Wählen Sie das Instance-Profil mit dem Namen `AWSPCS-getstarted-role`
  - Subnetze — Wählen Sie das Subnetz aus, mit dem der Name beginnt. `hpc-networking:PublicSubnetA` Anmeldeknoten werden in einem öffentlichen Subnetz ausgeführt, sodass Sie sie erreichen können, um sich anzumelden und Ihre Arbeit einzureichen.
  - Instanzen — Wählen Sie`c6i.xlarge`.
  - Skalierungskonfiguration — Geben Sie 1 für Min. Instanzanzahl den Wert ein. Geben 1 Sie für Max. Instanzanzahl den Wert ein. Wenn Sie beide Zähler auf setzen, läuft 1 immer ein einziger Anmeldeknoten als stabiler Einstiegspunkt in den Cluster.
- Geben Sie unter Zusätzliche Einstellungen Folgendes an:
  - AMI-ID — Wählen Sie ein AMI aus, das Sie verwenden möchten und das einen Namen im folgenden Format hat:

    ```
    aws-pcs-sample_ami-al2023-platform-slurm-version
    ```

> **Note**
>
> Beispiel-AMIs für Slurm 25.05 und frühere Versionen verwenden Amazon Linux (`2amzn2`) anstelle von Amazon Linux 2023 (). `al2023`

Weitere Informationen zu den Beispiel-AMIs finden Sie unter. Verwenden von Amazon Machine Images (AMIs) als Beispiel mit AWS STCK.

- Fügen Sie im Abschnitt „Aktionen für den Knotenlebenszyklus“ Skripts hinzu, die gemeinsam genutzten Speicher bereitstellen und Knotenprotokolle weiterleiten. Wählen Sie für jede der folgenden Aktionen die Option Skript hinzufügen aus. Skripte werden innerhalb einer Phase von oben nach unten ausgeführt. Fügen Sie sie daher in der angegebenen Reihenfolge hinzu. Weitere Informationen zu den von AWS verwalteten Skripten finden Sie unterAWS-gepflegte Skripte.

  1. configure-cloudwatch-logs — Leitet die Lebenszyklus-Aktionsprotokolle jedes Knotens an Amazon Logs weiter. CloudWatch Konfigurieren Sie es zuerst so, dass die Ausgabe der nachfolgenden Aktionen vom ersten Start des Knotens an erfasst wird.
     - Lifecycle-Phase — Wählen Sie`nodeBootstrapped`.
     - Speicherort des Skripts — Geben Sie `s3://aws-pcs-repo-`*`region`*`/aws-pcs-node-lifecycle-scripts/configure-cloudwatch-logs-v1-latest.sh` ihn ein und *`region`* ersetzen Sie ihn durch die AWS Region Ihres Clusters.
     - Name — Geben Sie ein`configure-cloudwatch-logs`.
     - Argumente — Lassen Sie dieses Feld leer. Das Skript sendet automatisch Protokolle an die `/aws/pcs/`*`cluster-id`*`/lifecycle` Protokollgruppe.
     - Verhalten bei der Fehlerbehandlung — Wählen Sie`CONTINUE`.
     - Ausführungsrichtlinie — Wählen Sie`FIRST_BOOT_ONLY`.
  2. configure-efs-homes — Mounten Sie das Amazon EFS-Dateisystem als Basisverzeichnis `/home` und konfiguriert den Knoten so, dass bei der ersten Anmeldung das Basisverzeichnis jedes Benutzers erstellt wird.
     - Lifecycle-Phase — Wählen Sie. `nodeBootstrapped`
     - Speicherort des Skripts — Geben Sie `s3://aws-pcs-repo-`*`region`*`/aws-pcs-node-lifecycle-scripts/configure-efs-homes-v1-latest.sh` ihn ein und *`region`* ersetzen Sie ihn durch die AWS Region Ihres Clusters.
     - Name — Geben Sie ein`configure-efs-homes`.

- Argumente — Geben Sie ein `--efs-id` *`efs-file-system-id`* `--home-base /home --options tls` und *`efs-file-system-id`* ersetzen Sie es durch die ID des EFS-Dateisystems, das Sie zuvor im Tutorial erstellt haben.
- Verhalten bei der Fehlerbehandlung — Wählen Sie`CONTINUE`.
- Ausführungsrichtlinie — Wählen Sie`EVERY_BOOT`.

3. mount-fsx-lustre — Mounten Sie das FSx for Lustre-Dateisystem unter für leistungsstarken gemeinsamen Scratch-Speicher. `/shared`
   - `nodeBootstrapped`Lifecycle-Phase — Wählen Sie aus.
   - Speicherort des Skripts — Geben Sie `s3://aws-pcs-repo-`*`region`*`/aws-pcs-node-lifecycle-scripts/mount-fsx-lustre-v1-latest.sh` ihn ein und *`region`* ersetzen Sie ihn durch die AWS Region Ihres Clusters (z. B.`us-east-1`).
   - Name — Geben Sie ein`mount-fsx-lustre`.
   - Argumente — Geben Sie ein `--fsx-dns-name` *`fsx-dns-name`* `--mount-name` *`mount-name`* `--mount-point /shared` und *`fsx-dns-name`* ersetzen Sie es durch den DNS-Namen und *`mount-name`* durch den Mount-Namen, den Sie bei der Erstellung des FSx for Lustre-Dateisystems notiert haben.
   - Verhalten bei der Fehlerbehandlung — Wählen Sie. `CONTINUE`
   - Ausführungsrichtlinie — Wählen Sie`EVERY_BOOT`.
4. set-shared-dir-mode — Setzt den Modus `/shared` auf weltweit beschreibbare, feste Zugriffsrechte (Modus`1777`), sodass jeder Benutzer darin Dateien erstellen kann. Es wird danach ausgeführt, `mount-fsx-lustre` sodass es für das gemountete Dateisystem gilt.
   - Lifecycle-Phase — Wählen Sie`nodeBootstrapped`.
   - Speicherort des Skripts — Geben Sie ein`https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/pcs-scripts/open_shared_dir/assets/set-shared-dir-mode-v1.0.0.sh`. Dieses Community-Skript wird von HPC Recipes for AWS veröffentlicht.
   - Name — Geben Sie ein`set-shared-dir-mode`.
   - Argumente — Geben Sie ein`--path /shared --mode 1777`.
   - Verhalten bei der Fehlerbehandlung — Wählen Sie`CONTINUE`.
   - Ausführungsrichtlinie — Wählen Sie`EVERY_BOOT`.

- Wählen Sie Compute-Knotengruppe erstellen aus.

Im Feld Status wird der Wert Creating angezeigt, während die Compute-Knotengruppe bereitgestellt wird. Sie können mit dem nächsten Schritt des Tutorials fortfahren, während es in Bearbeitung ist.

Erstellen Sie eine Rechenknotengruppe für die Ausführung von Rechenaufträgen in AWS STCK

In diesem Schritt starten Sie eine Rechenknotengruppe, die elastisch skaliert wird, um Jobs auszuführen, die an den Cluster gesendet werden.

Um die Compute-Knotengruppe zu erstellen

- Öffnen Sie die AWS PCS-Konsole und navigieren Sie zu Clusters.
- Wählen Sie den Cluster mit dem Namen `get-started`
- Navigieren Sie zu Compute Node Groups und wählen Sie Create aus.
- Geben Sie im Abschnitt zur Einrichtung der Compute-Knotengruppe Folgendes an:
  - Name der Compute-Knotengruppe — Geben Sie ein`compute-1`.
- Geben Sie unter Computerkonfiguration die folgenden Werte ein oder wählen Sie sie aus:
  - EC2-Startvorlage — Wählen Sie die Startvorlage aus, die den Namen enthält `compute-getstarted-lt`
  - IAM-Instanzprofil — Wählen Sie das Instance-Profil mit dem Namen `AWSPCS-getstarted-role`
  - Subnetze — Wählen Sie das Subnetz aus, mit dem der Name beginnt. `hpc-networking:PrivateSubnetA` Rechenknoten werden in einem privaten Subnetz ausgeführt, da sie Jobs verarbeiten und nicht über das Internet erreichbar sein müssen.
  - Instanzen — Wählen Sie aus`c6i.xlarge`.
  - Skalierungskonfiguration — Geben Sie `0` für Min. Instanzanzahl den Wert ein. Geben `4` Sie für Max. Instanzanzahl den Wert ein. Bei einer Mindestanzahl `0` von Werten kann AWS PCS die Gruppe so herunterskalieren, dass keine Instanzen mehr ausgeführt werden, sodass Sie nur dann für die Berechnung zahlen, wenn sich die Arbeit in der Warteschlange befindet.
- Geben Sie unter Zusätzliche Einstellungen Folgendes an:
  - AMI-ID — Wählen Sie ein AMI aus, das Sie verwenden möchten und das einen Namen im folgenden Format hat:

    ```
    aws-pcs-sample_ami-al2023-platform-slurm-version
    ```

> **Note**
>
> Beispiel-AMIs für Slurm 25.05 und frühere Versionen verwenden Amazon Linux (`2amzn2`) anstelle von Amazon Linux 2023 (). `al2023`

Weitere Informationen zu den Beispiel-AMIs finden Sie unter. Verwenden von Amazon Machine Images (AMIs) als Beispiel mit AWS STCK.

- Fügen Sie im Abschnitt „Aktionen für den Knotenlebenszyklus“ Skripts hinzu, die gemeinsam genutzten Speicher bereitstellen und Knotenprotokolle weiterleiten. Wählen Sie für jede der folgenden Aktionen die Option Skript hinzufügen aus. Skripte werden innerhalb einer Phase von oben nach unten ausgeführt. Fügen Sie sie daher in der angegebenen Reihenfolge hinzu. Weitere Informationen zu den von AWS verwalteten Skripten finden Sie unterAWS-gepflegte Skripte.

  1. configure-cloudwatch-logs — Leitet die Lebenszyklus-Aktionsprotokolle jedes Knotens an Amazon Logs weiter. CloudWatch Konfigurieren Sie es zuerst so, dass die Ausgabe der nachfolgenden Aktionen vom ersten Start des Knotens an erfasst wird.
     - Lifecycle-Phase — Wählen Sie`nodeBootstrapped`.
     - Speicherort des Skripts — Geben Sie `s3://aws-pcs-repo-`*`region`*`/aws-pcs-node-lifecycle-scripts/configure-cloudwatch-logs-v1-latest.sh` ihn ein und *`region`* ersetzen Sie ihn durch die AWS Region Ihres Clusters.
     - Name — Geben Sie ein`configure-cloudwatch-logs`.
     - Argumente — Lassen Sie dieses Feld leer. Das Skript sendet automatisch Protokolle an die `/aws/pcs/`*`cluster-id`*`/lifecycle` Protokollgruppe.
     - Verhalten bei der Fehlerbehandlung — Wählen Sie`CONTINUE`.
     - Ausführungsrichtlinie — Wählen Sie`FIRST_BOOT_ONLY`.
  2. configure-efs-homes — Mounten Sie das Amazon EFS-Dateisystem als Basisverzeichnis `/home` und konfiguriert den Knoten so, dass bei der ersten Anmeldung das Basisverzeichnis jedes Benutzers erstellt wird.
     - Lifecycle-Phase — Wählen Sie. `nodeBootstrapped`
     - Speicherort des Skripts — Geben Sie `s3://aws-pcs-repo-`*`region`*`/aws-pcs-node-lifecycle-scripts/configure-efs-homes-v1-latest.sh` ihn ein und *`region`* ersetzen Sie ihn durch die AWS Region Ihres Clusters.
     - Name — Geben Sie ein`configure-efs-homes`.

- Argumente — Geben Sie ein `--efs-id` *`efs-file-system-id`* `--home-base /home --options tls` und *`efs-file-system-id`* ersetzen Sie es durch die ID des EFS-Dateisystems, das Sie zuvor im Tutorial erstellt haben.
- Verhalten bei der Fehlerbehandlung — Wählen Sie`CONTINUE`.
- Ausführungsrichtlinie — Wählen Sie`EVERY_BOOT`.

3. mount-fsx-lustre — Mounten Sie das FSx for Lustre-Dateisystem unter für leistungsstarken gemeinsamen Scratch-Speicher. `/shared`
   - `nodeBootstrapped`Lifecycle-Phase — Wählen Sie aus.
   - Speicherort des Skripts — Geben Sie `s3://aws-pcs-repo-`*`region`*`/aws-pcs-node-lifecycle-scripts/mount-fsx-lustre-v1-latest.sh` ihn ein und *`region`* ersetzen Sie ihn durch die AWS Region Ihres Clusters (z. B.`us-east-1`).
   - Name — Geben Sie ein`mount-fsx-lustre`.
   - Argumente — Geben Sie ein `--fsx-dns-name` *`fsx-dns-name`* `--mount-name` *`mount-name`* `--mount-point /shared` und *`fsx-dns-name`* ersetzen Sie es durch den DNS-Namen und *`mount-name`* durch den Mount-Namen, den Sie bei der Erstellung des FSx for Lustre-Dateisystems notiert haben.
   - Verhalten bei der Fehlerbehandlung — Wählen Sie. `CONTINUE`
   - Ausführungsrichtlinie — Wählen Sie`EVERY_BOOT`.
4. set-shared-dir-mode — Setzt den Modus `/shared` auf weltweit beschreibbare, feste Zugriffsrechte (Modus`1777`), sodass jeder Benutzer darin Dateien erstellen kann. Es wird danach ausgeführt, `mount-fsx-lustre` sodass es für das gemountete Dateisystem gilt.
   - Lifecycle-Phase — Wählen Sie`nodeBootstrapped`.
   - Speicherort des Skripts — Geben Sie ein`https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/pcs-scripts/open_shared_dir/assets/set-shared-dir-mode-v1.0.0.sh`. Dieses Community-Skript wird von HPC Recipes for AWS veröffentlicht.
   - Name — Geben Sie ein`set-shared-dir-mode`.
   - Argumente — Geben Sie ein`--path /shared --mode 1777`.
   - Verhalten bei der Fehlerbehandlung — Wählen Sie`CONTINUE`.
   - Ausführungsrichtlinie — Wählen Sie`EVERY_BOOT`.

- Wählen Sie Compute-Knotengruppe erstellen aus.

Im Feld Status wird der Wert Creating angezeigt, während die Compute-Knotengruppe bereitgestellt wird.

> **Important**
>
> Warten Sie, bis im Statusfeld Aktiv angezeigt wird, bevor Sie mit dem nächsten Schritt in diesem Tutorial fortfahren.

Erstellen Sie eine Warteschlange zur Verwaltung von Jobs in AWS PCS

Sie reichen einen Job an eine Warteschlange weiter, um ihn auszuführen. Der Job verbleibt in der Warteschlange, bis AWS PCS die Ausführung auf einer Rechenknotengruppe plant. Jede Warteschlange ist einer oder mehreren Rechenknotengruppen zugeordnet, die die für die Verarbeitung erforderlichen EC2 Instanzen bereitstellen.

In diesem Schritt erstellen Sie eine Warteschlange, die die Rechenknotengruppe zur Verarbeitung von Jobs verwendet.

So erstellen Sie eine Warteschlange

- Öffnen Sie die AWS PCS-Konsole.
- Wählen Sie den genannten Cluster aus`get-started`.
- Navigieren Sie zu Compute Node Groups und stellen Sie sicher, dass der Status der `compute-1` Gruppe Aktiv lautet.

  > **Important**
  >
  > Der Status der `compute-1` Gruppe muss Aktiv sein, bevor Sie mit dem nächsten Schritt fortfahren können.

- Navigieren Sie zu Warteschlangen und wählen Sie Warteschlange erstellen.
  - Geben Sie im Abschnitt Warteschlangenkonfiguration die folgenden Werte an:
    - Name der Warteschlange — Geben Sie Folgendes ein: `demo`
    - Compute-Knotengruppen — Wählen Sie die benannte Compute-Knotengruppe aus`compute-1`.

- Wählen Sie Create queue (Warteschlange erstellen) aus.

Während die Warteschlange erstellt wird, wird im Statusfeld Creating angezeigt.

> **Important**
>
> Warten Sie, bis im Statusfeld Aktiv angezeigt wird, bevor Sie mit dem nächsten Schritt in diesem Tutorial fortfahren.

Connect zu Ihrem AWS PCS-Cluster her

Wenn der Status der `login` Compute-Knotengruppe Aktiv lautet, können Sie eine Verbindung zu der von ihr erstellten EC2 Instanz herstellen.

Um eine Verbindung zum Login-Knoten herzustellen

- Öffnen Sie die AWS PCS-Konsole und navigieren Sie zu Clusters.
- Wählen Sie den genannten Cluster aus`get-started`.
- Wählen Sie Compute Node Groups aus.
- Navigieren Sie zu der genannten Compute-Knotengruppe`login`.
- Suchen Sie die Compute-Knotengruppen-ID.
- Öffnen Sie in einem anderen Browserfenster oder einer anderen Registerkarte die EC2 Amazon-Konsole.
  - Wählen Sie Instances.
  - Suchen Sie nach EC2 Instances mit dem folgenden Tag. *`node-group-id`*Ersetzen Sie ihn durch den Wert der Compute-Knotengruppen-ID aus dem vorherigen Schritt. Es sollte 1 Instanz geben.

    ```
    aws:pcs:compute-node-group-id=node-group-id
    ```

  - Connect zur EC2 Instanz her. Sie können Session Manager oder SSH verwenden.

    Session Manager

    - Wählen Sie die Instance aus.
    - Wählen Sie Connect aus.
    - Wählen Sie unter Mit Instanz verbinden die Option Session Manager aus.

- Wählen Sie Connect aus.
- Wählen Sie Connect aus. In Ihrem Browser wird ein interaktives Terminal gestartet.

SSH

- Wählen Sie die Instance aus.
- Wählen Sie Connect aus.
- Wählen Sie unter Mit Instanz verbinden die Option SSH-Client aus.
- Folgen Sie den Anweisungen der Konsole.

> **Note**
>
> Der Benutzername für die Instanz ist **`ec2-user`**nicht`root`.

Erkunden Sie die Cluster-Umgebung in AWS PCS

Nachdem Sie sich beim Cluster angemeldet haben, können Sie Shell-Befehle ausführen. Sie können beispielsweise Benutzer wechseln, mit Daten auf gemeinsam genutzten Dateisystemen arbeiten und mit Slurm interagieren.

Benutzer ändern

Wenn Sie sich mit Session Manager beim Cluster angemeldet haben, sind Sie möglicherweise verbunden als`ssm-user`. Dies ist ein spezieller Benutzer, der für Session Manager erstellt wurde. Wechseln Sie mit dem folgenden Befehl zum Standardbenutzer auf Amazon Linux 2023. Sie müssen dies nicht tun, wenn Sie eine Verbindung über SSH hergestellt haben.

```
sudo su - ec2-user
```

Arbeiten Sie mit gemeinsam genutzten Dateisystemen

Mit dem Befehl können Sie überprüfen, ob das EFS-Dateisystem und die Dateisysteme FSx for Lustre verfügbar sind. `df -h` Die Ausgabe auf Ihrem Cluster sollte wie folgt aussehen:

```
[ec2-user@ip-10-3-6-103 ~]$ df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  3.8G     0  3.8G   0% /dev
tmpfs                     3.9G     0  3.9G   0% /dev/shm
```

```
tmpfs                       3.9G  556K  3.9G   1% /run
tmpfs                       3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/nvme0n1p1               24G   18G  6.6G  73% /
127.0.0.1:/                 8.0E     0  8.0E   0% /home
10.3.132.79@tcp:/zlshxbev   1.2T  7.5M  1.2T   1% /shared
tmpfs                       780M     0  780M   0% /run/user/0
tmpfs                       780M     0  780M   0% /run/user/1000
```

Das /home Dateisystem mountet 127.0.0.1 und hat eine sehr große Kapazität. Dies ist das EFS-Dateisystem, das Sie zu Beginn des Tutorials erstellt haben. Alle hier geschriebenen Dateien sind /home auf allen Knoten im Cluster unter verfügbar.

Das `/shared` Dateisystem mountet eine private IP und hat eine Kapazität von 1,2 TB. Dies ist das FSx for Lustre-Dateisystem, das Sie zu Beginn des Tutorials erstellt haben. Alle hier geschriebenen Dateien sind `/shared` auf allen Knoten im Cluster unter verfügbar.

Interagiere mit Slurm

Topics

- Listet Warteschlangen und Knoten auf
- Jobs anzeigen

Listet Warteschlangen und Knoten auf

Sie können die Warteschlangen und die Knoten, mit denen sie verknüpft sind, auflisten. `sinfo` Die Ausgabe Ihres Clusters sollte wie folgt aussehen:

```
[ec2-user@ip-10-3-6-103 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
demo         up   infinite      4  idle~ compute-1-[1-4]
[ec2-user@ip-10-3-6-103 ~]$
```

Notieren Sie sich die benannte Partitiondemo. Ihr Status ist up und sie hat maximal 4 Knoten. Es ist Knoten in der `compute-1` Knotengruppe zugeordnet. Wenn Sie die Compute-Knotengruppe bearbeiten und die maximale Anzahl von Instanzen auf 8 erhöhen, würde die Anzahl der Knoten lesen 8 und die Knotenliste würde lesen`compute-1-[1-8]`. Wenn Sie eine zweite Rechenknotengruppe `test` mit dem Namen 4 Knoten erstellen und sie der demo Warteschlange hinzufügen würden, würden diese Knoten auch in der Knotenliste angezeigt.

Jobs anzeigen

Sie können alle Jobs in jedem Status auf dem System mit auflistensqueue. Die Ausgabe Ihres Clusters sollte wie folgt aussehen:

```
[ec2-user@ip-10-3-6-103 ~]$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Versuchen Sie es später squeue erneut, wenn ein Slurm-Job aussteht oder läuft.

Führen Sie einen Einzelknotenjob in AWS PCS aus

Um einen Job mit Slurm auszuführen, bereiten Sie ein Einreichungsskript vor, in dem die Jobanforderungen angegeben sind, und senden es mit dem sbatch Befehl an eine Warteschlange. In der Regel erfolgt dies von einem gemeinsam genutzten Verzeichnis aus, sodass die Anmelde- und Rechenknoten über einen gemeinsamen Bereich für den Zugriff auf Dateien verfügen.

Connect zum Login-Knoten Ihres Clusters her und führen Sie die folgenden Befehle an der Shell-Eingabeaufforderung aus.

- Werden Sie der Standardbenutzer. Wechseln Sie in das gemeinsam genutzte Verzeichnis.

```
sudo su - ec2-user
cd /shared
```

- Verwenden Sie die folgenden Befehle, um ein Beispiel-Jobskript zu erstellen:

```
cat << EOF > job.sh
#!/bin/bash
#SBATCH -J single
#SBATCH -o single.%j.out
#SBATCH -e single.%j.err

echo "This is job \${SLURM_JOB_NAME} [\${SLURM_JOB_ID}] running on \
${SLURMD_NODENAME}, submitted from \${SLURM_SUBMIT_HOST}" && sleep 60 && echo "Job
 complete"
EOF
```

- Senden Sie das Jobskript an den Slurm-Scheduler:

```
sbatch -p demo job.sh
```

- Wenn der Job eingereicht wird, wird eine Job-ID als Zahl zurückgegeben. Verwenden Sie diese ID, um den Jobstatus zu überprüfen. Ersetzen Sie *job-id* den folgenden Befehl durch die Zahl, die von zurückgegeben wurde`sbatch`.

  ```
  squeue --job job-id
  ```

  Example

  ```
  squeue --job 1
  ```

  Der `squeue` Befehl gibt eine Ausgabe zurück, die der folgenden ähnelt:

  ```
  JOBID PARTITION NAME USER     ST TIME NODES NODELIST(REASON)
  1     demo      test ec2-user CF 0:47 1     compute-1
  ```

- Überprüfen Sie weiterhin den Status des Jobs, bis er den Status `R` (läuft) erreicht. Der Job ist erledigt, wenn `squeue` nichts zurückgegeben wird.
- Untersuchen Sie den Inhalt des `/shared` Verzeichnisses.

  ```
  ls -alth /shared
  ```

  Die Befehlsausgabe ähnelt der folgenden:

  ```
  -rw-rw-r- 1 ec2-user ec2-user 107 Mar 19 18:33 single.1.out
  -rw-rw-r- 1 ec2-user ec2-user 0 Mar 19 18:32 single.1.err
  -rw-rw-r- 1 ec2-user ec2-user 381 Mar 19 18:29 job.sh
  ```

  Die Dateien sind benannt `single.1.out` und `single.1.err` wurden von einem der Rechenknoten Ihres Clusters geschrieben. Da der Job in einem gemeinsam genutzten Verzeichnis (`/shared`) ausgeführt wurde, sind sie auch auf Ihrem Anmeldeknoten verfügbar. Aus diesem Grund haben Sie für diesen Cluster ein FSx For Lustre-Dateisystem konfiguriert.

- Untersuchen Sie den Inhalt der `single.1.out` Datei.

  ```
  cat /shared/single.1.out
  ```

Die Ausgabe sieht folgendermaßen oder ähnlich aus:

```
This is job test [1] running on compute-1, submitted from ip-10-3-13-181
Job complete
```

Führen Sie einen MPI-Job mit mehreren Knoten mit Slurm in PCS aus AWS

Diese Anweisungen demonstrieren die Verwendung von Slurm zur Ausführung eines MPI-Jobs (Message Passing Interface) in PCS. AWS

Führen Sie die folgenden Befehle an einer Shell-Eingabeaufforderung Ihres Login-Knotens aus.

- Werden Sie der Standardbenutzer. Wechseln Sie in sein Home-Verzeichnis.

```
sudo su - ec2-user
cd ~/
```

- Erstellen Sie Quellcode in der Programmiersprache C.

```
cat > hello.c << EOF
// * mpi-hello-world - https://www.mpitutorial.com
// Released under MIT License
//
// Copyright (c) 2014 MPI Tutorial.
//
// Permission is hereby granted, free of charge, to any person obtaining a copy
// of this software and associated documentation files (the "Software"), to
// deal in the Software without restriction, including without limitation the
// rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
// sell copies of the Software, and to permit persons to whom the Software is
// furnished to do so, subject to the following conditions:
// The above copyright notice and this permission notice shall be included in
// all copies or substantial portions of the Software.
//
// THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
// IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
// FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
// AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
// LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
```

```
// FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
// DEALINGS IN THE SOFTWARE.

#include <mpi.h>
#include <stdio.h>
#include <stddef.h>

int main(int argc, char** argv) {
  // Initialize the MPI environment. The two arguments to MPI Init are not
  // currently used by MPI implementations, but are there in case future
  // implementations might need the arguments.
  MPI_Init(NULL, NULL);

  // Get the number of processes
  int world_size;
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);

  // Get the rank of the process
  int world_rank;
  MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

  // Get the name of the processor
  char processor_name[MPI_MAX_PROCESSOR_NAME];
  int name_len;
  MPI_Get_processor_name(processor_name, &name_len);

  // Print off a hello world message
  printf("Hello world from processor %s, rank %d out of %d processors\n",
         processor_name, world_rank, world_size);

  // Finalize the MPI environment. No more MPI calls can be made after this
  MPI_Finalize();
}
EOF
```

- Laden Sie das OpenMPI-Modul.

```
module load openmpi
```

- Kompilieren Sie das C-Programm.

```
mpicc -o hello hello.c
```

- Schreiben Sie ein Slurm-Job-Skript.

```
cat > hello.sh << EOF
#!/bin/bash
#SBATCH -J multi
#SBATCH -o multi.out
#SBATCH -e multi.err
#SBATCH --exclusive
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=1

srun $HOME/hello
EOF
```

- Wechseln Sie in das gemeinsam genutzte Verzeichnis.

```
cd /shared
```

- Reichen Sie das Jobskript ein.

```
sbatch -p demo ~/hello.sh
```

- Wird verwendet`squeue`, um den Job zu überwachen, bis er erledigt ist.
- Überprüfen Sie den Inhalt von`multi.out`:

```
cat multi.out
```

Die Ausgabe sieht folgendermaßen oder ähnlich aus. Beachten Sie, dass jeder Rang seine eigene IP-Adresse hat, da er auf einem anderen Knoten lief.

```
Hello world from processor ip-10-3-133-204, rank 0 out of 4 processors
Hello world from processor ip-10-3-128-219, rank 2 out of 4 processors
Hello world from processor ip-10-3-141-26, rank 3 out of 4 processors
Hello world from processor ip-10-3-143-52, rank 1 out of 4 processor
```

Lösche deine AWS Ressourcen für AWS STCK

Nachdem Sie mit den Cluster- und Knotengruppen fertig sind, die Sie für dieses Tutorial erstellt haben, sollten Sie die von Ihnen erstellten Ressourcen löschen.

> **Important**
>
> Es fallen Abrechnungsgebühren für alle Ressourcen an, die in Ihrem AWS-Konto

Um zu löschen AWS PCS-Ressourcen, die Sie für dieses Tutorial erstellt haben

- Öffnen Sie die AWS PCS-Konsole.
- Navigieren Sie zu dem Cluster mit dem Namen get-started.
- Navigieren Sie zum Abschnitt Warteschlangen.
- Wählen Sie die Warteschlange mit dem Namen demo aus.
- Wählen Sie Löschen aus.

> **Important**
>
> Warten Sie, bis die Warteschlange gelöscht wurde, bevor Sie fortfahren.

- Navigieren Sie zum Abschnitt Compute-Knotengruppen.
- Wählen Sie die Compute-Knotengruppe mit dem Namen compute-1 aus.
- Wählen Sie Löschen aus.
- Wählen Sie die Rechenknotengruppe namens login aus.
- Wählen Sie Löschen aus.

> **Important**
>
> Warten Sie, bis beide Compute-Knotengruppen gelöscht wurden, bevor Sie fortfahren.

- Wählen Sie auf der Cluster-Detailseite für die ersten Schritte die Option Löschen aus.

> **Important**
>
> Warten Sie, bis der Cluster gelöscht wurde, bevor Sie mit den nachfolgenden Schritten fortfahren.

Um andere zu löschen AWS Ressourcen, die Sie für dieses Tutorial erstellt haben

- Öffnen Sie die IAM-Konsole.
  - Wählen Sie Roles.
  - Wählen Sie die benannte Rolle aus und klicken Sie AWSPCS-getstarted-role dann auf Löschen. Durch das Löschen der Rolle werden die AWS verwalteten Richtlinien von ihr getrennt. Die verwalteten Richtlinien selbst werden nicht gelöscht, da sie AWS Eigentümer sind und verwaltet werden.
- Öffnen Sie die CloudFormation -Konsole.
  - Wählen Sie den Stack mit dem Namen getstarted-lt aus.
  - Wählen Sie Löschen aus.

> **Important**
>
> Warten Sie, bis der Stack gelöscht ist, bevor Sie fortfahren.

- Öffnen Sie die Amazon-ECS-Konsole.
  - Wählen Sie Dateisysteme aus.
  - Wählen Sie das Dateisystem mit dem Namen getstarted-efs aus.
  - Wählen Sie Löschen aus.

> **Important**
>
> Warten Sie, bis das Dateisystem gelöscht ist, bevor Sie fortfahren.

- Öffnen Sie die Amazon FSx-Konsole.
  - Wählen Sie Dateisysteme aus.
  - Wählen Sie das Dateisystem mit dem Namen getstarted-fsx aus.
  - Wählen Sie Löschen aus.

> **Important**
>
> Warten Sie, bis das Dateisystem gelöscht ist, bevor Sie fortfahren.

- Öffnen Sie die CloudFormation -Konsole.
  - Wählen Sie den Stack mit dem Namen getstarted-sg aus.

- Wählen Sie Löschen aus.

- Öffnen Sie die CloudFormation -Konsole.
  - Wählen Sie den Stack mit dem Namen hpc-networking aus.
  - Wählen Sie Löschen aus.

Erste Schritte mit CloudFormation AWS PCS

Sie können es verwenden AWS CloudFormation , um einen AWS PCS-Cluster zu erstellen. CloudFormation ermöglicht es Ihnen, AWS Infrastrukturbereitstellungen vorhersehbar und wiederholt zu erstellen und bereitzustellen. Sie können CloudFormation die automatische Bereitstellung von Ressourcen aus vielen AWS Diensten verwenden, um äußerst zuverlässige, skalierbare und kostengünstige Anwendungen zu erstellen, AWS Cloud ohne die zugrunde liegende Infrastruktur erstellen und konfigurieren zu müssen. AWS CloudFormation ermöglicht es Ihnen, mithilfe einer Vorlagendatei eine Sammlung von Ressourcen zu einer einzigen Einheit, einem sogenannten Stapel, zu erstellen und zu löschen. Weitere Informationen zu CloudFormation finden Sie unter Was ist CloudFormation? im AWS CloudFormation Benutzerhandbuch. Weitere Informationen zu AWS PCS-Ressourcentypen finden Sie in CloudFormation der Referenz zu AWS PCS-Ressourcentypen im AWS CloudFormation Benutzerhandbuch.

Themen

- Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster
- Stellen Sie eine Connect zu einem AWS PCS-Cluster her, der erstellt wurde mit CloudFormation
- Bereinigen Sie einen AWS PCS-Cluster in CloudFormation
- Teile einer CloudFormation Vorlage für AWS 5 STCK
- CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster

Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster

Das folgende Verfahren verwendet eine CloudFormation Vorlage in der AWS-Managementkonsole um eine Probe zu erstellen AWS PCS-Cluster. Für weitere Informationen über CloudFormation, siehe Was ist CloudFormation? in der AWS CloudFormation Benutzerleitfaden. Für weitere Informationen über AWS PCS-Ressourcentypen in CloudFormation, siehe AWS Referenz zum PCS-Ressourcentyp in der AWS CloudFormation Benutzerleitfaden.

Um den Beispielcluster zu erstellen

1. Wählen Sie den AWS-Region um den Cluster zu erstellen in (der Link öffnet den CloudFormation Konsole mit der Vorlage):

- USA Ost (Nord-Virginia) (us-east-1)
- USA Ost (Ohio) (us-east-2)
- USA West (Oregon) (us-west-2)
- Afrika (Kapstadt) (af-south-1)
- Asien-Pazifik (Mumbai) (ap-south-1)
- Asien-Pazifik (Hyderabad) (ap-south-2)
- Asien-Pazifik (Singapur) (ap-southeast-1)
- Asien-Pazifik (Sydney) (ap-southeast-2)
- Asien-Pazifik (Jakarta) (ap-southeast-3)
- Asien-Pazifik (Melbourne) (ap-southeast-4)
- Asien-Pazifik (Malaysia) (ap-southeast-5)
- Asien-Pazifik (Neuseeland) (ap-southeast-6)
- Asien-Pazifik (Tokio) (ap-northeast-1)
- Asien-Pazifik (Seoul) (ap-northeast-2)
- Asien-Pazifik (Osaka) (ap-northeast-3)
- Kanada (Zentral) (ca-central-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Paris) (eu-west-3)
- Europa (Mailand) (eu-Süd-1)
- Europa (Spanien) (eu-Süd-2)
- Europa (Stockholm) (eu-nord-1)
- Südamerika (São Paulo) (sa-east-1)
- AWS GovCloud (US-East) (US-Regierung Ost-1)
- AWS GovCloud (US-West) (US-Regierung West-1)

2. Unter Geben Sie einen Stack-Namen an, geben Sie einen beschreibenden Namen ein. Das ist der Name für deinen CloudFormation stapeln. Die Vorlage verwendet diesen Wert als Namen für Ihren AWS PCS-Cluster.
3. Unter Parameter:

a. Unter SlurmVersion, wähle die Version von Slurm, die dein Cluster verwenden soll.

b. Unter NodeArchitecture, wähle x86 um einen Cluster bereitzustellen, der x86_64-kompatible Instanzen verwendet, oder wählen Sie Graviton um Arm64-Instanzen zu verwenden.

c. Für KeyName, wählen Sie ein SSH-Schlüsselpaar, um auf die Cluster-Anmeldeknoten zuzugreifen. Stellen Sie sicher, dass Sie über die PEM-Datei für das von Ihnen gewählte Schlüsselpaar verfügen.

d. Für ClientIpCidr, geben Sie einen IP-Bereich im CIDR-Format ein, um den Zugriff auf die Anmeldeknoten zu kontrollieren.

> ⚠ Warning
>
> Der Standardwert von `0.0.0.0/0` ermöglicht den Zugriff von allen IP-Adressen aus.

e. Belassen Sie die Werte für HpcRecipesS3Bucket und HpcRecipesBranch als ihre Standardwerte.

4. Unter Funktionen und Transformationen:

a. Markieren Sie das Kästchen, um das zu bestätigen CloudFormation erstellt IAM-Ressourcen.

b. Markieren Sie das Kontrollkästchen, um dies zu bestätigen CloudFormation erstellt IAM-Ressourcen mit benutzerdefinierten Namen.

c. Wählen Sie das Kontrollkästchen aus, um zu bestätigen `CAPABILITY_AUTO_EXPAND` für den neuen Stapel. Weitere Informationen finden Sie unter CreateStack in der AWS CloudFormation -API-Referenz.

5. Wählen Sie Stack erstellen aus.

6. Überwachen Sie den Status Ihres Stacks. Sie können eine Verbindung zum Cluster herstellen, nachdem der Status des Stacks lautet `CREATE_COMPLETE`.

Stellen Sie eine Connect zu einem AWS PCS-Cluster her, der erstellt wurde mit CloudFormation

Nachdem Sie einen AWS PCS-Cluster anhand einer CloudFormation Vorlage erstellt haben, können Sie den Cluster mit der AWS PCS-Konsole (im AWS-Managementkonsole) verwalten. Sie können

auch eine Verbindung zu einem der Anmeldeknoten des Clusters herstellen, um den Cluster zu verwalten, Jobs auszuführen und Daten zu verwalten. Der CloudFormation Stack bietet Links, über die Sie eine Verbindung zu Ihrem Cluster herstellen können.

Um eine Verbindung zu Ihrem Cluster herzustellen

1. Öffnen Sie die CloudFormation -Konsole.
2. Wählen Sie den Stack aus, den Sie erstellt haben.
3. Wählen Sie die Registerkarte Ausgaben des Stacks.

   Der Stapel bietet die folgenden Links:

   - PcsConsoleUrl— Wählen Sie diesen Link, um die AWS PCS-Konsole mit dem ausgewählten Cluster zu öffnen. Sie können ihn verwenden, um die Cluster-, Knotengruppen- und Warteschlangenkonfigurationen zu erkunden.
   - Ec2 ConsoleUrl — Wählen Sie diesen Link, um die Amazon EC2 EC2-Konsole zu öffnen, die so gefiltert ist, dass die Instances angezeigt werden, die von der Login-Knotengruppe des Clusters verwaltet werden.

     In dieser Ansicht können Sie eine Instanz auswählen und Connect wählen. Die Instanz des Beispielclusters unterstützt eingehendes SSH und AWS Systems Manager Verbindungen in einem Webbrowser. Weitere Informationen finden Sie unter Connect zu Ihrem AWS PCS-Cluster her.

     Nachdem Sie eine Verbindung zu einer Anmeldeinstanz hergestellt haben, können Sie dem Tutorial unter folgen. Erkunden Sie die Cluster-Umgebung in AWS PCS

Bereinigen Sie einen AWS PCS-Cluster in CloudFormation

Wenn Sie früher CloudFormation Ihren AWS PCS-Cluster erstellt haben, können Sie die CloudFormation Konsole öffnen und den Stack löschen, um den Cluster und alle zugehörigen Ressourcen zu löschen.

> **Important**
>
> Wenn Sie für den Beispielcluster zusätzliche Compute-Knotengruppen oder Warteschlangen in Ihrem Cluster erstellt haben (zusätzlich zu den `login` `compute-1` Gruppen, die mit der CloudFormation Beispielvorlage erstellt wurden), müssen Sie die AWS PCS-Konsole

verwenden oder AWS CLI diese Ressourcen löschen, bevor Sie den CloudFormation Stack löschen. Weitere Informationen finden Sie unter Löschen eines Clusters in AWS PCS.

Teile einer CloudFormation Vorlage für AWS 5 STCK

Eine CloudFormation Vorlage besteht aus einem oder mehreren Abschnitten, die jeweils einem bestimmten Zweck dienen. CloudFormation definiert Standardformat, Syntax und Sprache in einer Vorlage. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Arbeiten mit CloudFormation Vorlagen.

CloudFormation Vorlagen sind in hohem Maße anpassbar und daher können ihre Formate variieren. Um die für die Erstellung eines AWS PCS-Clusters erforderlichen Teile einer CloudFormation Vorlage zu verstehen, empfehlen wir Ihnen, sich die Beispielvorlage anzusehen, die wir für die Erstellung eines Beispielclusters bereitstellen. In diesem Thema werden die Abschnitte dieser Beispielvorlage kurz erläutert.

Important

Die Codebeispiele in diesem Thema sind nicht vollständig. Das Vorhandensein von Auslassungspunkten (`[...]`) weist darauf hin, dass zusätzlicher Code vorhanden ist, der nicht angezeigt wird. Informationen zum Herunterladen der vollständigen YAML-formatted CloudFormation Vorlage finden Sie unterCloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.

Inhalt

- Header
- Metadaten
- Parameters
- Mappings
- Ressourcen
- Outputs

Header

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: AWS Parallel Computing Service "getting started" cluster
```

`AWSTemplateFormatVersion`identifiziert die Version des Vorlagenformats, der die Vorlage entspricht. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter Syntax der Version des CloudFormation Vorlagenformats.

`Transform`gibt ein Makro an, das zur Verarbeitung der Vorlage CloudFormation verwendet wird. Weitere Informationen finden Sie im Abschnitt zum Transformieren von CloudFormation Vorlagen im AWS CloudFormation Benutzerhandbuch. Die `AWS::Serverless-2016-10-31` Transformation ermöglicht CloudFormation die Verarbeitung einer in der AWS Serverless Application Model (AWS SAM) -Syntax geschriebenen Vorlage. Weitere Informationen finden Sie unter `AWS::Serverless` transform im AWS CloudFormation Benutzerhandbuch.

Metadaten

```
### Stack metadata
Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: PCS Cluster configuration
        Parameters:
          - SlurmVersion
          - ManagedAccounting
          - AccountingPolicyEnforcement
      - Label:
          default: PCS ComputeNodeGroups configuration
        Parameters:
          - NodeArchitecture
          - KeyName
          - ClientIpCidr
      - Label:
          default: HPC Recipes configuration
        Parameters:
          - HpcRecipesS3Bucket
          - HpcRecipesBranch
```

Der `metadata` Abschnitt einer CloudFormation Vorlage enthält Informationen über die Vorlage selbst. Mit der Beispielvorlage wird ein vollständiger HPC-Cluster (High Performance Computing) erstellt, der AWS PCS verwendet. Im Metadatenbereich der Beispielvorlage werden Parameter deklariert, die steuern, wie der entsprechende CloudFormation Stack gestartet (bereitgestellt) wird. Es gibt Parameter, die die Architekturauswahl (`NodeArchitecture`), die Slurm-Version (`SlurmVersion`) und die Zugriffskontrollen (`KeyNameandClientIpCidr`) steuern.

Parameters

Der `Parameters` Abschnitt definiert die benutzerdefinierten Parameter für die Vorlage. CloudFormation verwendet diese Parameterdefinitionen, um das Formular zu erstellen und zu validieren, mit dem Sie interagieren, wenn Sie einen Stapel von dieser Vorlage aus starten.

```
Parameters:

  NodeArchitecture:
    Type: String
    Default: x86
    AllowedValues:
      - x86
      - Graviton
    Description: Processor architecture for the login and compute node instances

  SlurmVersion:
    Type: String
    Default: 25.11
    Description: Version of Slurm to use
    AllowedValues:
         - 25.05
         - 25.11

  ManagedAccounting:
    Type: String
    Default: 'disabled'
    AllowedValues:
      - 'enabled'
      - 'disabled'
    Description: Monitor cluster usage, manage access control, and enforce resource
 limits with Slurm accounting.

  AccountingPolicyEnforcement:
    Description: Specify which Slurm accounting policies to enforce
```

```
    Type: String
    Default: none
    AllowedValues:
      - none
      - 'associations,limits,safe'

  KeyName:
    Description: SSH keypair to log in to the head node
    Type: AWS::EC2::KeyPair::KeyName
    AllowedPattern: ".+"  # Required

  ClientIpCidr:
    Description: IP(s) allowed to access the login node over SSH. We recommend that
you restrict it with your own IP/subnet (x.x.x.x/32 for your own ip or x.x.x.x/24 for
range. Replace x.x.x.x with your own PUBLIC IP. You can get your public IP using tools
such as https://ifconfig.co/)
    Default: 127.0.0.1/32
    Type: String
    AllowedPattern: (\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})
    ConstraintDescription: Value must be a valid IP or network range of the form
x.x.x.x/x.

  HpcRecipesS3Bucket:
    Type: String
    Default: aws-hpc-recipes
    Description: HPC Recipes for AWS S3 bucket
    AllowedValues:
         - aws-hpc-recipes
         - aws-hpc-recipes-dev
  HpcRecipesBranch:
    Type: String
    Default: main
    Description: HPC Recipes for AWS release branch
    AllowedPattern: '^(?!.*/\.git$)(?!.*/\.)(?!.*\\.\.)[a-zA-Z0-9-_\.]+$'
```

Mappings

Der `Mappings` Abschnitt definiert Schlüssel-Wert-Paare, die Werte angeben, die auf bestimmten Bedingungen oder Abhängigkeiten basieren.

```
Mappings:

  Architecture:
```

```
    AmiArchParameter:
      Graviton: arm64
      x86: x86_64
    LoginNodeInstances:
      Graviton: c7g.xlarge
      x86: c6i.xlarge
    ComputeNodeInstances:
      Graviton: c7g.xlarge
      x86: c6i.xlarge
```

Ressourcen

Der `Resources` Abschnitt deklariert die AWS Ressourcen, die bereitgestellt und konfiguriert werden müssen, als Teil des Stacks.

```
Resources:

  [...]
```

Die Vorlage stellt die Beispiel-Cluster-Infrastruktur in Schichten bereit. Sie beginnt mit „`Networking`Für die VPC-Konfiguration“. Der Speicher wird von zwei Systemen bereitgestellt: `EfsStorage` für gemeinsam genutzten Speicher und `FSxLStorage` für Hochleistungsspeicher. Der Core-Cluster wird eingerichtet durch`PCSCluster`.

```

  Networking:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        ProvisionSubnetsC: "False"
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/net/hpc_large_scale/assets/main.yaml'

  EfsStorage:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        SubnetIds: !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        SubnetCount: 1
        VpcId: !GetAtt [ Networking, Outputs.VPC ]
```

```
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/storage/efs_simple/assets/main.yaml'

  FSxLStorage:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        PerUnitStorageThroughput: 125
        SubnetId: !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        VpcId: !GetAtt [ Networking, Outputs.VPC ]
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/storage/fsx_lustre/assets/persistent.yaml'

  [...]

  # Cluster
  PCSCluster:
    Type: AWS::PCS::Cluster
    Properties:
      Name: !Sub '${AWS::StackName}'
      Size: SMALL
      Scheduler:
        Type: SLURM
        Version: !Ref SlurmVersion
      Networking:
        SubnetIds:
          - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
        SecurityGroupIds:
          - !GetAtt [ PCSSecurityGroup, Outputs.ClusterSecurityGroupId ]
```

Für Rechenressourcen erstellt die Vorlage zwei Knotengruppen: `PCSNodeGroupLogin` für einen einzelnen Anmeldeknoten und `PCSNodeGroupCompute` für bis zu vier Rechenknoten. Diese Knotengruppen werden von `PCSInstanceProfile` für Berechtigungen und `PCSLaunchTemplate` für Instanzkonfigurationen unterstützt.

```
  # Compute Node groups
  PCSInstanceProfile:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        # We have to regionalize this in case CX use the template in more than one
 region. Otherwise,
```

```
        # the create action will fail since instance-role-${AWS::StackName} already
 exists!
        RoleName: !Sub '${AWS::StackName}-${AWS::Region}'
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/pcs-iip-minimal.yaml'

  PCSLaunchTemplate:
    Type: AWS::CloudFormation::Stack
    Properties:
      Parameters:
        VpcDefaultSecurityGroupId: !GetAtt [ Networking, Outputs.SecurityGroup ]
        ClusterSecurityGroupId: !GetAtt [ PCSSecurityGroup,
 Outputs.ClusterSecurityGroupId ]
        SshSecurityGroupId: !GetAtt [ PCSSecurityGroup,
 Outputs.InboundSshSecurityGroupId ]
        EfsFilesystemSecurityGroupId: !GetAtt [ EfsStorage, Outputs.SecurityGroupId ]
        FSxLustreFilesystemSecurityGroupId: !GetAtt [ FSxLStorage,
 Outputs.FSxLustreSecurityGroupId ]
        SshKeyName: !Ref KeyName
      TemplateURL: !Sub 'https://${HpcRecipesS3Bucket}.s3.amazonaws.com/
${HpcRecipesBranch}/recipes/pcs/getting_started/assets/cfn-pcs-lt-efs-fsxl.yaml'

  # Compute Node groups - Login Nodes
  PCSNodeGroupLogin:
    Type: AWS::PCS::ComputeNodeGroup
    Properties:
      ClusterId: !GetAtt [PCSCluster, Id]
      Name: login
      ScalingConfiguration:
        MinInstanceCount: 1
        MaxInstanceCount: 1
      IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
      CustomLaunchTemplate:
        TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.LoginLaunchTemplateId ]
        Version: 1
      SubnetIds:
        - !GetAtt [ Networking, Outputs.DefaultPublicSubnet ]
      AmiId: !GetAtt [PcsSampleAmi, AmiId]
      InstanceConfigs:
        - InstanceType: !FindInMap [ Architecture, LoginNodeInstances, !Ref
 NodeArchitecture ]

  # Compute Node groups - Compute Nodes
  PCSNodeGroupCompute:
```

```
    Type: AWS::PCS::ComputeNodeGroup
    Properties:
      ClusterId: !GetAtt [PCSCluster, Id]
      Name: compute-1
      ScalingConfiguration:
        MinInstanceCount: 0
        MaxInstanceCount: 4
      IamInstanceProfileArn: !GetAtt [ PCSInstanceProfile, Outputs.InstanceProfileArn ]
      CustomLaunchTemplate:
        TemplateId: !GetAtt [ PCSLaunchTemplate, Outputs.ComputeLaunchTemplateId ]
        Version: 1
      SubnetIds:
        - !GetAtt [ Networking, Outputs.DefaultPrivateSubnet ]
      AmiId: !GetAtt [PcsSampleAmi, AmiId]
      InstanceConfigs:
        - InstanceType: !FindInMap [ Architecture, ComputeNodeInstances, !Ref
NodeArchitecture ]
```

Beide Knotengruppen enthalten einen `NodeLifecycleActions` Block, der gemeinsam genutzten Speicher bereitstellt und Knotenprotokolle in der Phase an Amazon CloudWatch Logs weiterleitet. `nodeBootstrapped` Dies ersetzt den älteren Ansatz, Dateisysteme mithilfe von Launch-Template-Benutzerdaten einzubinden. Amazon EFS ist `/home` für Benutzerstartverzeichnisse gemountet, und FSx for Lustre ist für Hochleistungs-Scratch-Speicher gemountet`/shared`. Eine vierte Aktion leitet die Aktionsprotokolle des Knotenlebenszyklus an Amazon Logs weiter. CloudWatch Die Mount- und Directory-Aktionen werden verwendet, `EVERY_BOOT` sodass sie bei jedem Neustart ausgeführt werden und Speicherplatz verfügbar bleibt, während sie `configure-cloudwatch-logs` verwendet werden, `FIRST_BOOT_ONLY` weil die CloudWatch Agentenkonfiguration auch nach Neustarts erhalten bleibt und nicht erneut angewendet werden muss. Weitere Informationen zu Aktionen im Lebenszyklus von Knoten finden Sie unter. Aktionen im Lebenszyklus eines Knotens Einzelheiten zu den Skripten, die AWS verwaltet werden, finden Sie unterAWS-gepflegte Skripte.

```
    NodeLifecycleActions:
      ScriptCachingPolicy: CACHE_ONCE
      Stages:
        NodeBootstrapped:
          # Forward node lifecycle action logs to CloudWatch Logs. Runs first so the
          # actions that follow have their output captured from the node's first
boot.
          - Name: configure-cloudwatch-logs
```

```
              ScriptSource:
                ScriptLocation: !Sub 's3://aws-pcs-repo-${AWS::Region}/aws-pcs-node-
lifecycle-scripts/configure-cloudwatch-logs-v1-latest.sh'
              OnError: CONTINUE
              ExecutionPolicy: FIRST_BOOT_ONLY
            # Mount EFS as the home-directory base at /home and create home dirs on
 first login.
            - Name: configure-efs-homes
              ScriptSource:
                ScriptLocation: !Sub 's3://aws-pcs-repo-${AWS::Region}/aws-pcs-node-
lifecycle-scripts/configure-efs-homes-v1-latest.sh'
              Arguments:
                - '--efs-id'
                - !GetAtt [ EfsStorage, Outputs.EFSFilesystemId ]
                - '--home-base'
                - '/home'
                - '--options'
                - 'tls'
              OnError: CONTINUE
              ExecutionPolicy: EVERY_BOOT
            # Mount FSx for Lustre at /shared for high-performance shared scratch
 storage.
            - Name: mount-fsx-lustre
              ScriptSource:
                ScriptLocation: !Sub 's3://aws-pcs-repo-${AWS::Region}/aws-pcs-node-
lifecycle-scripts/mount-fsx-lustre-v1-latest.sh'
              Arguments:
                - '--fsx-dns-name'
                - !GetAtt FSxLustreLookup.DNSName
                - '--mount-name'
                - !GetAtt FSxLustreLookup.MountName
                - '--mount-point'
                - '/shared'
              OnError: CONTINUE
              ExecutionPolicy: EVERY_BOOT
            # Set /shared to world-writable sticky mode (1777). Runs after mount-fsx-
lustre
            # so it applies to the mounted file system, not an empty directory.
            - Name: set-shared-dir-mode
              ScriptSource:
                ScriptLocation: !Sub 'https://${HpcRecipesS3Bucket}.s3.us-
east-1.amazonaws.com/${HpcRecipesBranch}/recipes/pcs-scripts/open_shared_dir/assets/
set-shared-dir-mode-v1.0.0.sh'
              Arguments:
```

```
              - '--path'
              - '/shared'
              - '--mode'
              - '1777'
            OnError: CONTINUE
            ExecutionPolicy: EVERY_BOOT
```

Die Auftragsplanung wird über abgewickeltPCSQueueCompute.

```
PCSQueueCompute:
  Type: AWS::PCS::Queue
  Properties:
    ClusterId: !GetAtt [PCSCluster, Id]
    Name: demo
    ComputeNodeGroupConfigurations:
      - ComputeNodeGroupId: !GetAtt [PCSNodeGroupCompute, Id]
```

Die AMI-Auswahl erfolgt automatisch über die PcsAMILookupFn Lambda-Funktion und die zugehörigen Ressourcen.

```
PcsAMILookupRole:
  Type: AWS::IAM::Role
  [...]

PcsAMILookupFn:
  Type: AWS::Lambda::Function
  Properties:
    Runtime: python3.12
    Handler: index.handler
    Role: !GetAtt PcsAMILookupRole.Arn
    Code:
      [...]
    Timeout: 30
    MemorySize: 128

# Example of using the custom resource to look up an AMI
PcsSampleAmi:
  Type: Custom::AMILookup
  Properties:
    ServiceToken: !GetAtt PcsAMILookupFn.Arn
    OperatingSystem: 'amzn2'
```

```
      Architecture: !FindInMap [ Architecture, AmiArchParameter, !Ref
 NodeArchitecture ]
      SlurmVersion: !Ref SlurmVersion
```

Outputs

Die Vorlage gibt Cluster-Identifikations- und Verwaltungs-URLs über `ClusterIdPcsConsoleUrl`, und `Ec2ConsoleUrl` aus.

```
Outputs:
  ClusterId:
    Description: The Id of the PCS cluster
    Value: !GetAtt [ PCSCluster, Id ]

  PcsConsoleUrl:
    Description: URL to access the cluster in the PCS console
    Value: !Sub
      - https://${ConsoleDomain}/pcs/home?region=${AWS::Region}#/clusters/${ClusterId}
      - { ConsoleDomain: !If [ GovCloud, 'console.amazonaws-us-gov.com', !If [ China,
 'console.amazonaws.cn', !Sub '${AWS::Region}.console.aws.amazon.com']],
          ClusterId: !GetAtt [ PCSCluster, Id ]
        }
    Export:
      Name: !Sub ${AWS::StackName}-PcsConsoleUrl

  Ec2ConsoleUrl:
    Description: URL to access instance(s) in the login node group via Session Manager
    Value: !Sub
      - https://${ConsoleDomain}/ec2/home?region=
${AWS::Region}#Instances:instanceState=running;tag:aws:pcs:compute-node-group-id=
${NodeGroupLoginId}
      - { ConsoleDomain: !If [ GovCloud, 'console.amazonaws-us-gov.com', !If [ China,
 'console.amazonaws.cn', !Sub '${AWS::Region}.console.aws.amazon.com']],
          NodeGroupLoginId: !GetAtt [ PCSNodeGroupLogin, Id ]
        }
    Export:
      Name: !Sub ${AWS::StackName}-Ec2ConsoleUrl
```

CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster

AWS-Region Name	AWS-Region	Quelltext anzeigen	Stack starten
USA Ost (Nord-Virginia)	us-east-1	Laden Sie YAML herunter	
USA Ost (Ohio)	us-east-2	Laden Sie YAML herunter	
USA West (Oregon)	us-west-2	Laden Sie YAML herunter	
Afrika (Kapstadt)	af-south-1	Laden Sie YAML herunter	
Asien-Pazifik (Mumbai)	ap-south-1	Laden Sie YAML herunter	
Asien-Pazifik (Hyderabad)	ap-south-2	Laden Sie YAML herunter	
Asien-Pazifik (Singapur)	ap-southeast-1	Laden Sie YAML herunter	
Asien-Pazifik (Sydney)	ap-southeast-2	Laden Sie YAML herunter	
Asien-Pazifik (Jakarta)	ap-southeast-3	Laden Sie YAML herunter	
Asien-Pazifik (Melbourne)	ap-southeast-4	Laden Sie YAML herunter	
Asien-Pazifik (Malaysia)	ap-southeast-5	Laden Sie YAML herunter	

AWS-Region Name	AWS-Region	Quelltext anzeigen	Stack starten
Asien-Pazifik (Neuseeland)	ap-southeast-6	Laden Sie YAML herunter	Launch Stack
Asien-Pazifik (Tokio)	ap-northeast-1	Laden Sie YAML herunter	Launch Stack
Asien-Pazifik (Seoul)	ap-northeast-2	Laden Sie YAML herunter	Launch Stack
Asia Pacific (Osaka)	ap-northeast-3	Laden Sie YAML herunter	Launch Stack
Kanada (Zentral)	ca-central-1	Laden Sie YAML herunter	Launch Stack
Europa (Frankfurt)	eu-central-1	Laden Sie YAML herunter	Launch Stack
Europa (Irland)	eu-west-1	Laden Sie YAML herunter	Launch Stack
Europa (London)	eu-west-2	Laden Sie YAML herunter	Launch Stack
Europa (Paris)	eu-west-3	Laden Sie YAML herunter	Launch Stack
Europa (Milan)	eu-south-1	Laden Sie YAML herunter	Launch Stack
Europa (Spain)	eu-south-2	Laden Sie YAML herunter	Launch Stack
Europa (Stockholm)	eu-north-1	Laden Sie YAML herunter	Launch Stack
Südamerika (São Paulo)	sa-east-1	Laden Sie YAML herunter	Launch Stack

AWS-Region Name	AWS-Region	Quelltext anzeigen	Stack starten
AWS GovCloud (US-East)	us-gov-east-1	Laden Sie YAML herunter	
AWS GovCloud (US-West)	us-gov-west-1	Laden Sie YAML herunter	

AWS PCS-Cluster

Ein AWS PCS-Cluster besteht aus den folgenden Komponenten:

- Verwaltete Instanzen der HPC System Scheduler-Software, wie z. B. der Slurm Control Daemon (). `slurmctld`
- Komponenten, die sich in den HPC-Systemplaner integrieren lassen, um EC2 Amazon-Instances bereitzustellen und zu verwalten.
- Komponenten, die in den HPC-Systemplaner integriert sind, um Protokolle und Metriken an Amazon zu übertragen. CloudWatch

Diese Komponenten werden in einem Konto ausgeführt, das von verwaltet wird. AWS Sie arbeiten zusammen, um EC2 Amazon-Instances in Ihrem Kundenkonto zu verwalten. AWS PCS stellt elastische Netzwerkschnittstellen in Ihrem Amazon VPC-Subnetz bereit, um Konnektivität von der Scheduler-Software zu EC2 Amazon-Instances bereitzustellen (z. B. um die Planung von Batch-Jobs auf diesen zu unterstützen und es Benutzern zu ermöglichen, Scheduler-Befehle auszuführen, um diese Jobs aufzulisten und zu verwalten).

Themen

- Einen Cluster erstellen in AWS STK.
- Aktualisierung eines Clusters in AWS STK.
- Löschen eines Clusters in AWS PCS
- Clustergröße in AWS PCS
- Arbeiten mit Cluster-Secrets in AWS STCK

Einen Cluster erstellen in AWS STK.

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was bei der Erstellung eines Clusters in AWS Parallel Computing Service (AWS PCS) zu beachten ist. Wenn Sie zum ersten Mal einen AWS PCS-Cluster erstellen, empfehlen wir Ihnen, wie folgt vorzugehenFangen Sie an mit AWS Dienst für parallele Datenverarbeitung. Das Tutorial kann Ihnen helfen, ein funktionierendes HPC-System zu erstellen, ohne auf alle verfügbaren Optionen und Systemarchitekturen eingehen zu müssen, die möglich sind.

> **Note**
>
> Nach der Erstellung eines Clusters können Sie viele Konfigurationseinstellungen ändern, ohne Ihre Infrastruktur neu aufbauen zu müssen. Weitere Informationen finden Sie unter Aktualisierung eines Clusters in AWS STK..

> **Note**
>
> Sie können benutzerdefinierte Slurm-Einstellungen konfigurieren, um erweiterte Planungsrichtlinien und Ressourcenmanagement zu implementieren. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK.

Voraussetzungen

- Eine bestehende VPC und ein Subnetz, die die Anforderungen erfüllenAWS PCS-Netzwerke. Bevor Sie einen Cluster für den Produktionseinsatz bereitstellen, empfehlen wir, dass Sie sich ein umfassendes Verständnis der VPC- und Subnetzanforderungen aneignen. Informationen zum Erstellen einer VPC und eines Subnetzes finden Sie unter. Erstellen Sie eine VPC für Ihre AWS PCS-Cluster
- Ein IAM-Prinzipal mit Berechtigungen zum Erstellen und Verwalten AWS von PCS-Ressourcen. Weitere Informationen finden Sie unter Identitäts- und Zugriffsmanagement für AWS Dienst für parallele Datenverarbeitung.

Erstellen Sie ein AWS PCS-Cluster

Sie können das AWS-Managementkonsole oder verwenden AWS CLI , um einen Cluster zu erstellen.

AWS-Managementkonsole

So erstellen Sie einen Cluster

1. Öffnen Sie die AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/home#/clusters und wählen Sie Create cluster aus.
2. Geben Sie im Abschnitt Cluster-Setup die folgenden Felder ein:

- Clustername — Ein Name für Ihren Cluster. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht länger als 40 Zeichen sein. Der Name muss innerhalb des AWS-Region und AWS-Konto , in dem Sie den Cluster erstellen, eindeutig sein.
- Scheduler — Wählen Sie einen Scheduler und eine Version aus. Weitere Informationen finden Sie unter Slurm-Versionen in AWS 5 STCK.
- Controller-Größe — Wählen Sie eine Größe für Ihren Controller. Dies bestimmt, wie viele gleichzeitige Jobs und Rechenknoten vom AWS PCS-Cluster verwaltet werden können. Sie können die Controller-Größe nur festlegen, wenn der Cluster erstellt wird. Weitere Informationen zur Größenbestimmung finden Sie unterClustergröße in AWS PCS.

3. Wählen Sie im Abschnitt Netzwerk Werte für die folgenden Felder aus:

- Netzwerktyp — Wählen Sie den IP-Adresstyp für Ihren Cluster. Ihr Cluster kann entweder IPv4 oder IPv6 verwenden, aber nicht beide. Die VPC und die Subnetze müssen denselben Netzwerkadresstyp verwenden. Der IP-Adressblock, den Sie für jedes Subnetz verwenden, muss mindestens eine verfügbare Adresse haben. AWS reserviert einige Adressen in jedem Subnetz. Weitere Informationen finden Sie unter Subnetz-CIDR-Blöcke im Amazon-VPC-Benutzerhandbuch.
- VPC — Wählen Sie eine vorhandene VPC, die die AWS PCS-Anforderungen erfüllt. Weitere Informationen finden Sie unter AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen. Nachdem Sie den Cluster erstellt haben, können Sie seine VPC nicht mehr ändern. Wenn keine VPCs aufgeführt sind, müssen Sie zuerst eine erstellen.
- Subnetz — Alle verfügbaren Subnetze in der ausgewählten VPC werden aufgelistet. Wählen Sie ein Subnetz, das die PCS-Subnetzanforderungen erfüllt. AWS Weitere Informationen finden Sie unter AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen. Wir empfehlen Ihnen, ein privates Subnetz auszuwählen, um zu verhindern, dass Ihre Scheduler-Endpunkte dem öffentlichen Internet ausgesetzt werden.
- Sicherheitsgruppen — Geben Sie die Sicherheitsgruppe (n) an, die AWS PCS den Netzwerkschnittstellen zuordnen soll, die es für Ihren Cluster erstellt. Sie müssen mindestens eine Sicherheitsgruppe auswählen, die die Kommunikation zwischen Ihrem Cluster und seinen Rechenknoten ermöglicht. Sie können Schnell eine Sicherheitsgruppe erstellen auswählen, damit AWS PCS eine Sicherheitsgruppe mit der erforderlichen Konfiguration in der ausgewählten VPC erstellt, oder Sie können eine vorhandene

Sicherheitsgruppe auswählen. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zur Sicherheitsgruppe.

4. (Optional) Im Abschnitt zur Konfiguration der Slurm-Buchhaltung können Sie die Slurm-Buchhaltung aktivieren und die Abrechnungsparameter festlegen. Weitere Informationen finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.
5. (Optional) Im Abschnitt zur Scheduler-Konfiguration können Sie Parameternamen- und Wertepaare hinzufügen, um zusätzliche Slurm-Einstellungen zu konfigurieren. Eine vollständige Liste der unterstützten Parameter finden Sie unter. Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster Sie können auch benutzerdefinierte Cgroup-Einstellungen (siheKonfiguration benutzerdefinierter Cgroup-Einstellungen in AWS STCK) und benutzerdefinierte SlurmDBD-Einstellungen (siehe) konfigurieren. Konfiguration benutzerdefinierter SlurmDBD-Einstellungen in AWS 5 STÜCK

   Darüber hinaus können Sie auch die folgenden Einstellungen angeben:

   - Scale-down Leerlaufzeit — (Optional) Die Zeit in Sekunden, bis ein Knoten im Leerlauf herunterskaliert wird. Dies gilt standardmäßig für alle Compute-Knotengruppen im Cluster. Für Slurm 25.11 oder höher können einzelne Compute-Knotengruppen diesen Wert mit ihrer eigenen Einstellung für die Scale-down Leerlaufzeit überschreiben.
6. (Optional) Fügen Sie unter Tags beliebige Tags zu Ihrem AWS PCS-Cluster hinzu.
7. Wählen Sie Cluster erstellen. Das Statusfeld wird angezeigt`Creating`, während der AWS PCS den Cluster erstellt. Dieser Vorgang kann einige Minuten dauern.

> **Important**
>
> AWS-Region Pro `Creating` Bundesstaat kann es nur einen Cluster geben AWS-Konto. AWS PCS gibt beim Versuch, einen Cluster zu erstellen, einen Fehler zurück, wenn sich bereits ein Cluster in einem `Creating` Status befindet.

AWS CLI

So erstellen Sie einen Cluster

1. Erstellen Sie den Cluster mit dem folgenden Befehl. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:

- *region*Ersetzen Sie es durch die ID des Clusters AWS-Region , in dem Sie Ihren Cluster erstellen möchten, z. `us-east-1` B.
- Ersetzen Sie *my-cluster* durch Ihren Cluster-Namen. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Sie muss mit einem alphabetischen Zeichen beginnen und darf nicht länger als 40 Zeichen sein. Der Name muss innerhalb des Clusters AWS-Region und an dem AWS-Konto Ort, an dem Sie den Cluster erstellen, eindeutig sein.
- *25.11*Ersetzen Sie es durch eine unterstützte Version von Slurm.

  > **Note**
  >
  > AWS PCS unterstützt derzeit Slurm 25.11 und 25.05.

- Ersetzen Sie durch eine beliebige *SMALL* unterstützte Clustergröße. Dies bestimmt, wie viele gleichzeitige Jobs und Rechenknoten vom AWS PCS-Cluster verwaltet werden können. Es kann nur festgelegt werden, wenn der Cluster erstellt wird. Weitere Informationen zur Dimensionierung finden Sie unterClustergröße in AWS PCS.
- Ersetzen Sie den Wert für `subnetIds` durch Ihren eigenen. Wir empfehlen Ihnen, ein privates Subnetz auszuwählen, um zu verhindern, dass Ihre Scheduler-Endpunkte dem öffentlichen Internet ausgesetzt werden.
- Geben Sie an`securityGroupIds`, welche Netzwerkschnittstellen AWS PCS den Netzwerkschnittstellen zuordnen soll, die es für Ihren Cluster erstellt. Die Sicherheitsgruppen müssen sich in derselben VPC wie der Cluster befinden. Sie müssen mindestens eine Sicherheitsgruppe auswählen, die die Kommunikation zwischen Ihrem Cluster und seinen Rechenknoten ermöglicht. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zur Sicherheitsgruppe.

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=25.11 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
```

- um IPv6 zu verwenden, fügen Sie `networkType=IPV6` es der `--networking` Konfiguration hinzu.

```
  --networking networkType=IPV6,subnetIds=subnet-
ExampleId1,securityGroupIds=sg-ExampleId1
```

- Optional können Sie die `--slurm-configration` Option hinzufügen, das Slurm-Verhalten anzupassen und die Slurm-Konfigurationsoptionen festzulegen. Im folgenden Beispiel wird die Leerlaufzeit beim Herunterfahren auf 60 Minuten (3600 Sekunden) festgelegt, die Slurm-Accounting-Funktion aktiviert und Einstellungen als Wert für angegebens`lurm.conf`. `slurmCustomSettings` Weitere Informationen finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.

  > **Note**
  >
  > Accounting wird für Slurm 24.11 oder höher unterstützt.

  > **Note**
  >
  > Die Cluster-Ebene `scaleDownIdleTimeInSeconds` gilt standardmäßig für alle Compute-Knotengruppen im Cluster. Für Slurm 25.11 oder höher können einzelne Compute-Knotengruppen diesen Wert mit ihrer eigenen Einstellung überschreiben. `scaleDownIdleTimeInSeconds`

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM,version=25.11 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1
    --slurm-configuration
 scaleDownIdleTimeInSeconds=3600,accounting='{mode=STANDARD}',slurmCustomSettings='[{p
```

2. Die Bereitstellung des Clusters kann mehrere Minuten dauern. Sie können den Status Ihres Clusters mit dem folgenden Befehl überprüfen. Fahren Sie erst mit der Erstellung von Warteschlangen oder Compute-Knotengruppen fort, wenn das Statusfeld des Clusters angezeigt wird`ACTIVE`.

```
aws pcs get-cluster --region region --cluster-identifier my-cluster
```

> **Important**
>
> AWS-Region Pro `Creating` AWS-Konto Bundesstaat kann es nur einen Cluster geben. AWS PCS gibt beim Versuch, einen Cluster zu erstellen, einen Fehler zurück, wenn sich bereits ein Cluster in einem `Creating` Status befindet.

Empfohlene nächste Schritte für Ihren Cluster

- Fügen Sie Compute-Knotengruppen hinzu.
- Fügen Sie Warteschlangen hinzu.
- Aktivieren Sie die Protokollierung.

Aktualisierung eines Clusters in AWS STK.

AWS Mit PCS können Sie Clusterkonfigurationen nach der Erstellung über die UpdateCluster API oder Konsole aktualisieren. Sie können die Cluster-Einstellungen ändern, ohne Ihre Infrastruktur neu aufbauen zu müssen, wodurch der Betriebsaufwand reduziert und Unterbrechungen minimiert werden.

Vorteile von Cluster-Updates

Durch die Aktualisierung von AWS PCS-Clustern können Sie die HPC-Infrastruktur ohne Betriebsunterbrechung an neue Anforderungen anpassen. Konfigurationsänderungen dauern Minuten statt der Stunden oder länger, die für die Neuerstellung von Clustern erforderlich sind. Diese Funktion ist wichtig für Produktionsumgebungen, in denen nur minimale Ausfallzeiten erforderlich sind, und für Teams, die die Clustereinstellungen anpassen müssen, wenn sich die Arbeitslastmuster ändern.

Unterstützte Konfigurationsänderungen

Sie können vier Hauptkategorien von Einstellungen ändern:

- Scheduler-Version — Aktualisieren Sie auf eine neuere unterstützte Scheduler-Version. Weitere Informationen finden Sie unter Aktualisierung der Scheduler-Version eines Clusters in AWS STCK.
- Kontoführungskonfiguration — Aktivieren oder deaktivieren Sie die verwaltete Buchhaltung und konfigurieren Sie die Aufbewahrungseinstellungen.
- Scale-down Verhalten — Passen Sie den `scaleDownIdleTime` Parameter an, der steuert, wie lange dynamische Instanzen inaktiv bleiben, bevor AWS PCS sie automatisch beendet.
- Benutzerdefinierte Slurm-Einstellungen — Ändern Sie alle unterstützten Slurm-Einstellungen, die auf Cluster-Ebene gelten, einschließlich Prolog, Epilog und. SelectTypeParameters

Einschränkungen

Sie können bestimmte Konfigurationen nach der Clustererstellung nicht ändern. Dazu zählen:

- Konfigurationen von Sicherheitsgruppen
- Auswahl des VPC-Subnetzes
- Cluster-Größe
- Clustername

Diese Einstellungen sind grundlegend für die Architektur des Clusters und erfordern die Erstellung eines neuen Clusters, um sie zu ändern. Informationen zum Aktualisieren der Scheduler-Version finden Sie unter. Aktualisierung der Scheduler-Version eines Clusters in AWS STCK

Voraussetzungen für Cluster-Updates

Stellen Sie vor dem Aktualisieren eines Clusters sicher, dass die folgenden Bedingungen erfüllt sind:

- Der Cluster muss sich im `SUSPENDED` Status `ACTIVEUPDATE_FAILED`, oder befinden
- Alle zugehörigen Ressourcen (Warteschlangen, Compute-Knotengruppen) müssen sich im Status befinden `ACTIVE`
- Sie müssen über die entsprechenden IAM-Berechtigungen für den Vorgang verfügen UpdateCluster
- Es können keine anderen Aktualisierungsvorgänge ausgeführt werden

Aktualisierungsprozess und Auswirkung auf den Job

Während eines Aktualisierungsvorgangs führen die Rechenknoten weiterhin bestehende Jobs aus, auch wenn der Cluster-Controller kurzzeitig nicht erreichbar ist. In diesem Zeitraum kann das System jedoch keine neuen Auftragseinreichungen annehmen oder Entscheidungen zur Terminplanung treffen.

Sie können Cluster-Updates sowohl über die Konsole als auch über die API-Schnittstelle überwachen. Der Cluster durchläuft während eines Updates die folgenden Zustände:

- `UPDATING`- Aktualisierung läuft
- `ACTIVE`- Das Update wurde erfolgreich abgeschlossen
- `UPDATE_FAILED`- Beim Update ist ein Fehler aufgetreten

Abrechnung bei Updates

Die standardmäßigen Stundengebühren für Ihren AWS PCS-Cluster werden während der Aktualisierungsvorgänge weiterhin berechnet. Wenn Sie einen Cluster aktualisieren, um die Kontoführung zu deaktivieren, wird die Abrechnung für die Abrechnungsfunktion beendet, sobald der Cluster den `UPDATING` Status erreicht. Wenn Sie die Kontoführung aktivieren, beginnt die Abrechnung erst, wenn der Cluster das Update erfolgreich abgeschlossen hat und in den `ACTIVE` Status zurückkehrt.

Themen

- Aktualisiere ein AWS PCS-Cluster
- Aktualisierung der Scheduler-Version eines Clusters in AWS STCK
- Häufig gestellte Fragen zum Aktualisieren von Clustern in AWS STCK
- Problembehandlung bei AWS PCS-Cluster-Updates

Aktualisiere ein AWS PCS-Cluster

Gehen Sie wie folgt vor, um die Scheduler-Einstellungen, die Accounting-Konfiguration und die benutzerdefinierten Slurm-Einstellungen auf Ihrem Cluster zu ändern. Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster.

Voraussetzungen

- Der Cluster muss sich im Status `ACTIVEUPDATE_FAILED`, oder befinden `SUSPENDED`
- Alle zugehörigen Ressourcen (Warteschlangen, Compute-Knotengruppen) müssen sich im Status befinden `ACTIVE`
- Es können keine anderen Aktualisierungsvorgänge ausgeführt werden

Verfahren

AWS-Managementkonsole

1. Öffnen Sie die AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den zu aktualisierenden Cluster aus.
4. Wählen Sie Bearbeiten aus.
5. Ändern Sie auf der Seite Cluster bearbeiten die gewünschten Einstellungen:
   - Wählen Sie unter Clusterdetails eine neue Scheduler-Version aus der Scheduler-Dropdown-Liste aus. Weitere Informationen finden Sie unter Aktualisierung der Scheduler-Version eines Clusters in AWS STCK.
   - Aktualisieren Sie unter Scheduler-Konfiguration die Scale-down Leerlaufzeit, um zu steuern, wie lange dynamische Instanzen inaktiv bleiben, bevor sie automatisch beendet werden.
   - Ändern Sie die Prolog -, Epilog - und Select-type Parametereinstellungen nach Bedarf.
   - Aktivieren, deaktivieren oder konfigurieren Sie die Aufbewahrungszeit für Managed Accounting.
   - Fügen Sie unter Zusätzliche Scheduler-Einstellungen benutzerdefinierte Slurm-Einstellungen hinzu, bearbeiten oder entfernen Sie sie. Weitere Informationen zu unterstützten Parametern finden Sie unter. Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster

> **Note**
>
> Felder, die nicht bearbeitet werden können, werden schreibgeschützt angezeigt und zeigen ihre aktuellen Werte an.

6. Wählen Sie „Aktualisieren“, um die Änderungen zu senden.
7. Überwachen Sie den Cluster-Status, der während des Vorgangs als „Aktualisierung“ angezeigt wird. Der Status ändert sich, wenn das Update erfolgreich abgeschlossen wurde.

AWS CLI

1. Öffnen Sie ein Terminal oder eine Befehlszeile.
2. Überprüfen Sie den Clusterstatus mit dem folgenden Befehl:

```
aws pcs get-cluster --cluster-identifier my-cluster
```

3. Senden Sie eine Aktualisierungsanfrage anhand eines der folgenden Beispiele:

   - So aktivieren Sie Managed Accounting:

   ```
   aws pcs update-cluster --cluster-identifier my-cluster \
   --slurm-configuration 'accounting={mode=STANDARD}'
   ```

   - Um eine Slurm Prolog-Einstellung zu aktualisieren:

   ```
   aws pcs update-cluster --cluster-identifier my-cluster \
   --slurm-configuration \
   'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/
   prolog.sh"}]'
   ```

   - Um die Leerlaufzeit beim Herunterskalieren zu aktualisieren:

   ```
   aws pcs update-cluster --cluster-identifier my-cluster \
   --slurm-configuration 'scaleDownIdleTimeInSeconds=300'
   ```

   - Um die Scheduler-Version zu aktualisieren:

   ```
   aws pcs update-cluster --cluster-identifier my-cluster \
   ```

```
--scheduler version=25.11
```

4. Überwachen Sie den Aktualisierungsfortschritt, indem Sie den Clusterstatus überprüfen:

```
aws pcs get-cluster --cluster-identifier my-cluster
```

Nach einer erfolgreichen Aktualisierungsanforderung gibt der Befehl das Cluster-Objekt mit allen Änderungen zurück. Der Clusterstatus ändert sich von `UPDATING` bis nach `ACTIVE` Abschluss.

Aktualisierung der Scheduler-Version eines Clusters in AWS STCK

AWS Mit PCS können Sie die Scheduler-Version auf einem vorhandenen Cluster aktualisieren, ohne Ihre Infrastruktur neu aufbauen zu müssen. Durch Versionsupdates wird der Cluster-Controller auf eine neuere Slurm-Hauptversion umgestellt, sodass Sie auf neue Funktionen, Leistungsverbesserungen und Sicherheitspatches zugreifen können. Neuere Versionen haben auch eine längere Lebensdauer des Supports, bevor sie das Ende ihrer Lebensdauer erreichen.

-Übersicht

Ein Scheduler-Versionsupdate umfasst drei Vorgänge:

1. Bereiten Sie Ziel-AMIs vor — Erstellen oder identifizieren Sie AMIs, die die Ziel-Slurm-Version und die neueste Version enthalten AWS PCS-Agent.
2. Aktualisieren Sie den Cluster (`UpdateCluster`) — Verschiebt den Controller auf die Ziel-Slurm-Hauptversion.
3. Compute-Knotengruppen aktualisieren (`UpdateComputeNodeGroup`) — Verweist jede Rechenknotengruppe auf ein neues AMI, sodass neue Knoten die Zielversion verwenden. Weitere Informationen finden Sie unter Aktualisierung einer AWS PCS-Rechenknotengruppe.

Es gibt zwei Pfade, denen Sie folgen können. Wählen Sie eine Option, die darauf basiert, ob Sie eine Arbeitsunterbrechung tolerieren können.

Unabhängig davon, für welche Option Sie sich entscheiden, müssen alle Knoten im Cluster vor dem Start des Prozesses dieselbe Version „A“ ausführen, und am Ende des Prozesses müssen alle Knoten dieselbe Version „B“ ausführen. Option 2 funktioniert unabhängig von Ihrer Clusterkonfiguration.

	Option 1: Fortlaufendes Update	Option 2: Full-fleet Wartungsstopp
Ausführen von Aufgaben	Erfordert keine Kündigung	Alle laufenden Jobs wurden beendet
AMI-Anforderung	Muss beinhalten beides aktuelle und Ziel-Slurm-Versionen	Benötigt nur die Ziel-Slurm-Version
Mindestversion des Cluster-Controllers	24.05	Keine Einschränkung
Rechne die Flotte nach dem Controller-Update	Vorübergehend gemischte Versionen ; Entleerungsschritte erforderlich. Wir empfehlen, den Zeitraum, in dem gemischte Versionen in einem Cluster verwendet werden, so gering wie möglich zu halten.	Alle Knoten starten in der Zielversion neu

> **Note**
>
> Stellen Sie vor dem Start sicher, dass alle Rechenknoten auf dem neuesten Patch von Slurm Version A und auf der neuesten Version sind AWS PCS-Agent.

Einschränkungen

- Wenn Ihr Cluster verwendet SPANK-Plugins (Plugstack-Konfiguration), AWS PCS unterstützt Option 1 nicht. Ein fortlaufendes Update kann zu einer Nichtübereinstimmung der Plugstack-Konfigurationsversion und zu Plugin-Fehlern führen. Diese Einschränkung gilt nicht, wenn du auf Slurm 26.05 oder höher aktualisierst, weil AWS PCS speichert die Plugstack-Konfiguration an einem Ort, der von der Slurm-Version unabhängig ist.
- Wenn Ihr Cluster verwendet Slurm-CLI-Filter-Plugins, AWS PCS unterstützt Option 1 nicht. Während eines fortlaufenden Updates können Rechenknoten, auf denen eine frühere Slurm-Version ausgeführt wird, die aktualisierte Cluster-Konfiguration möglicherweise nicht analysieren und starten möglicherweise nicht. Diese Einschränkung gilt nicht, wenn Sie von Slurm 25.11 oder

höher aktualisieren, weil AWS PCS speichert das Plugin-Skript an einem Ort, der von der Slurm-Version unabhängig ist.

Versionskompatibilität

In der folgenden Tabelle sind die unterstützten Zielversionen aufgeführt, auf die je nach aktueller Cluster-Version aktualisiert werden kann. Es wird immer empfohlen, ein Upgrade auf die neueste zulässige Version durchzuführen (fett dargestellt).

Der Cluster und alle Rechenknoten müssen immer dieselbe Slurm-Version ausführen, bevor ein Update initiiert wird.

Aktuelle Cluster-Version	Kompatible Zielversionen
26.05	Nicht zutreffend
25,11	26,05
25,05	26,05, 25,11
24.11 (EOL)	26,05, 25.11, 25.05
24,05 (EOL)	25,11, 25.05, 24.11
23.11 (EOL)	(nur über Option 2) 25.05, 24.11, 24.05

Weitere Informationen zu unterstützten Versionen und Enddaten finden Sie unter Slurm-Versionen in AWS 5 STCK.

Sie können nicht mehr als drei Hauptversionen in einem einzigen Update überspringen. Wenn Ihre Zielversion Ihrer aktuellen Version mehr als drei Hauptversionen voraus ist, führen Sie das Update in mehreren aufeinanderfolgenden Schritten durch. Ein Beispiel mit mehreren Schritten finden Sie unter Beispiel: Aktualisierung über mehrere Versionen hinweg.

Auswirkungen auf laufende Jobs

Während des Updates ist der Slurm-Controller kurzzeitig nicht verfügbar. Dies hat folgende Auswirkungen:

- Jobs werden ausgeführt — Bei Option 1 (fortlaufendes Update) werden Jobs, die bereits auf Rechenknoten ausgeführt werden, weiterhin ausgeführt. Für die Rechenknoten muss der Controller nicht für die aktive Auftragsausführung verfügbar sein. Bei Option 2 (vollständige Wartung der Flotte) werden alle laufenden Jobs beendet, wenn die Flotte heruntergefahren wird.
- Neue Stellenausschreibungen — Sie können keine neuen Jobs einreichen oder Scheduler-Befehle ausführen, solange der Controller nicht verfügbar ist.
- Skalierung — Die automatische Skalierung wird während des Updates unterbrochen. Bis zum Abschluss des Updates werden keine neuen Instances gestartet und keine Instances wegen Down-Scale-Down beendet.
- Daten zur Buchhaltung — Wenn die Buchhaltung aktiviert ist, bleiben die Buchhaltungsdaten während des gesamten Updates erhalten. In der Buchhaltungsdatenbank gespeicherte Auftragsdatensätze bleiben auch nach der Versionsänderung bestehen.
- Slurm REST-API — Wenn die Slurm-REST-API auf dem Cluster aktiviert ist, wird sie im Rahmen von automatisch auf die neue Scheduler-Version aktualisiert `UpdateCluster` Betrieb. Der REST-API-Endpunkt ist während des Updates nicht verfügbar und wird fortgesetzt, wenn der Cluster zu `ACTIVE` Bundesstaat. Weitere Informationen finden Sie unter Setzen Sie die REST-API ein AWS 5 STCK.

Kombinieren von Versionsupdates mit Konfigurationsänderungen

Sie können ein Versionsupdate mit anderen Konfigurationsänderungen in einem einzigen Schritt kombinieren `UpdateCluster` Anfrage. Beispielsweise können Sie die Scheduler-Version aktualisieren und die Abrechnung in derselben Operation aktivieren.

> **Note**
>
> Fügen Sie keine für die Zielversion spezifischen Slurm-Einstellungen hinzu, solange die Flotte noch Knoten der Vorgängerversion enthält. Die Konfiguration wird auf alle Knoten verteilt; auf die alten `slurmd` erkennt möglicherweise keine neuen Parameter.

Themen

- Aktualisieren Sie die Scheduler-Version eines AWS PCS-Cluster
- Fehlerbehebung AWS Aktualisierungen der PCS-Cluster-Versionen

Aktualisieren Sie die Scheduler-Version eines AWS PCS-Cluster

Gehen Sie wie folgt vor, um die Scheduler-Version auf Ihrem Cluster zu aktualisieren. Je nachdem, ob Sie eine Arbeitsunterbrechung tolerieren können, gibt es zwei Optionen. Weitere Informationen zur Auswahl zwischen Optionen finden Sie unter Aktualisierung der Scheduler-Version eines Clusters in AWS STCK.

> **Note**
>
> Wir empfehlen, dass Sie das neue AMI und das Aktualisierungsverfahren auf einem Cluster außerhalb der Produktionsumgebung testen, bevor Sie Änderungen an Ihrer Produktionsumgebung vornehmen.

Option 1: Fortlaufendes Update

Der Controller wird aktualisiert, während die Flotte weiterläuft. Bestehende Knoten verwenden weiterhin die vorherige Slurm-Version, bis sie entleert und ersetzt werden. Neue Knoten, die nach dem Update gestartet werden, verwenden die Zielversion. Laufende Jobs werden nicht unterbrochen.

Wann sollte man Folgendes verwenden:

- Der Cluster-Controller ist auf Slurm-Version 24.05 oder höher.
- Sie können AMIs bereitstellen, die sowohl die aktuelle als auch die Ziel-Slurm-Version enthalten.

Schritt 0 — Überprüfen Sie den Startstatus

Auf Ihrem Cluster wird Controller-Version „A“ (z. B. 24.11) ausgeführt, und Sie möchten auf Version „B“ (z. B. 25.11) migrieren. Vergewissern Sie sich, dass alle Rechenknoten in Ihrer Flotte dieselbe Hauptversion ausführen, indem Sie diesen Befehl von einem Cluster-Knoten aus verwenden:

```
sinfo -N -o "%N %T %v"

# Example output:
#   NODELIST    STATE VERSION
#   compute-1   idle  24.11.7
#   compute-2   idle  24.11.7
```

Bestätigen Sie AWS PCS-Agent-Version auf einem Rechenknoten. Stellen Sie mit Systems Manager eine Verbindung zum Knoten her und überprüfen Sie das Bootstrap-Protokoll:

```
grep "PCS Agent version" /var/log/amazon/pcs/bootstrap.log | tail -1

# Example output:
#   /opt/aws/pcs/bin/pcs_bootstrap_init.sh: INFO: Bootstrap starting with PCS Agent
 version: 1.3.2-1
```

Fortlaufende Updates sind erforderlich AWS PCS-Agent-Version 1.5.1-1 oder höher auf allen Compute-Node-AMIs. Weitere Informationen finden Sie unter AWS PCS-Agent-Versionen.

Schritt 1 — Vorbereitung und Einführung der AMIs mit zwei Versionen

Erstellen oder identifizieren Sie AMIs, die Folgendes beinhalten sowohl Slurm Version A als auch Version B und die neueste AWS PCS-Agent.

- Sie können das neueste verwenden PCS-ready DLA Mis. Solche AMIs werden mit den letzten drei unterstützten Slurm-Versionen ausgeliefert. Weitere Informationen finden Sie unter Verwenden von PCS-ready DLAMI mit AWS 5 STCK.
- Sie können eine bauen benutzerdefiniertes AMI, indem Sie den Installationsschritten für Slurm-Pakete folgen und AWS PCS-Agent. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.
- Wir empfehlen das nicht AWS PCS-Beispiel-AMI für den Produktionsgebrauch. Diese AMIs dienen nur zum Testen.

> **Note**
>
> Wenn Ihr AMI mehr als eine Slurm-Version enthält, AWS PCS wählt automatisch die Version aus, die zum Controller passt. Die Installation zusätzlicher Versionen verursacht keine Probleme.

Sobald die AMIs bereit sind:

1. Rufen Sie an `UpdateComputeNodeGroup` auf jeder Rechenknotengruppe, um das neue AMI mit zwei Versionen einzurichten. Die Knoten werden wie folgt in DRAIN versetzt AWS PCS und werden auf das neue AMI migrieren.

   AWS-Managementkonsole

   1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.

2. Wählen Sie im Navigationsbereich Cluster, und wählen Sie dann den Cluster aus.
3. Wählen Sie Knotengruppen berechnen, wählen Sie die zu aktualisierende Rechenknotengruppe aus, und wählen Sie Bearbeiten.
4. Unter Konfiguration berechnen, geben Sie die ID des AMI mit zwei Versionen in AMI-ID.
5. Wählen Sie Aktualisieren aus. Die Compute-Knotengruppe wird angezeigt als `UPDATING` während die Änderung angewendet wird.
6. Wiederholen Sie diese Schritte für jede Rechenknotengruppe im Cluster.

AWS CLI

Führen Sie den folgenden Befehl für jede Rechenknotengruppe im Cluster aus:

```
aws pcs update-compute-node-group \
  --cluster-identifier cluster-id \
  --compute-node-group-identifier cng-id \
  --ami-id new-ami-id
```

2. Warten Sie, bis entladene Knoten ihre Aufgaben abgeschlossen haben, beendet werden und durch Knoten ersetzt werden, die das neue AMI mit zwei Versionen verwenden. Stellen Sie sicher, dass alle EC2-Instances im Cluster das neue AMI verwenden, und zwar mit:

```
aws ec2 describe-instances \
    --filters "Name=tag:aws:pcs:cluster-id,Values=cluster-id" \
    --query "Reservations[].Instances[].[InstanceId,ImageId,State.Name]" \
    --output table
```

Schritt 2 — Aktualisieren Sie den Cluster-Controller

Rufen Sie an `UpdateCluster` mit `scheduler.version` auf Version B gesetzt

AWS-Managementkonsole

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, der aktualisiert werden soll, und wählen Sie Bearbeiten.
4. Unter Einzelheiten zum Cluster, wählen Sie die Ziel-Scheduler-Version aus Scheduler Dropdown.

5. Wählen Sie Aktualisieren um das Versionsupdate einzureichen.
6. Überwachen Sie den Cluster-Status. Der Cluster wird angezeigt als `UPDATING` während des Updates und kehrt zurück zu `ACTIVE` wenn abgeschlossen. Das Update ist in der Regel in 5—15 Minuten abgeschlossen.

AWS CLI

```
aws pcs update-cluster \
  --cluster-identifier cluster-id \
  --scheduler version=25.11
```

Warten Sie, bis der Cluster zu zurückgekehrt ist `ACTIVE`. Das Update ist in der Regel in 5—15 Minuten abgeschlossen.

Während dieses Vorgangs ist der Controller kurzzeitig nicht verfügbar:

- Jobs auf Rechenknoten ausführen mit der Ausführung fortfahren.
- Neue Jobübermittlungen und Scheduler-Befehle sind erst verfügbar, wenn das Update abgeschlossen ist.
- Die automatische Skalierung wird angehalten, bis der Cluster wieder `ACTIVE`.

Nach dem Update befindet sich die Computerflotte in einem gemischter Zustand: Knoten, die vor dem Update laufen, verwenden weiterhin die Slurm-Version A `slurmd`; neue Knoten verwenden Slurm-Version B. Dies wird erwartet.

> **Note**
>
> Fügen Sie keine Slurm-Einstellungen hinzu, die für Version B spezifisch sind, solange die Flotte noch Knoten in Version A enthält. Die Konfiguration wird an alle Knoten verteilt; die alten `slurmd` erkennt neue Parameter möglicherweise nicht.

Wenn der Cluster nicht zurückkehrt zu `ACTIVE` oder `UPDATE_FAILED` innerhalb von 30 Minuten, Kontakt AWS Unterstützung für Unterstützung.

Schritt 3 — Entwässern Sie Knoten, auf denen noch Slurm Version A läuft

Identifizieren und entleeren Sie Knoten, die noch in der Vorgängerversion installiert sind. Listen Sie von einem Knoten des Clusters aus jeden Knoten mit seinem Status und seiner Slurm-Version auf:

```
sinfo -N -o "%N %T %v"
```

Führen Sie für jeden Knoten, der immer noch Version A meldet, den folgenden Befehl aus. Verwenden Sie den Knotennamen aus dem `NODELIST` Spalte. Der EC2-Hostname ist kein gültiger Slurm-Knotenname.

```
scontrol update NodeName=node State=POWER_DOWN_ASAP Reason="Slurm version update"
```

Sobald entleerte Knoten ihre aktuellen Jobs beendet haben, werden sie beendet und durch Knoten in Slurm-Version B ersetzt.

Schritt 4 — Verifizieren Sie eine konsistente Flotte auf Slurm-Version B

Überwachen Sie die Flotte, bis alle Knoten Slurm-Version B melden. Führen Sie von einem Knoten des Clusters aus:

```
sinfo -N -o "%N %T %v"
```

Während die Knoten recycelt werden, passieren sie die `DRAINING`, `POWERING_DOWN`, und `POWERED_DOWN` Staaten. Ein Knoten in der `POWERED_DOWN` Zustand (dargestellt mit einem ~ Suffix, wie `idle~`) läuft nicht. Bis AWS PCS startet es erneut, es zeigt die Version an, die es zuletzt ausgeführt hat, oder `N/A` wenn es noch nie gestartet wurde. Das Update ist abgeschlossen, wenn keine Knoten mehr in einem `DRAIN`, `DRAINING`, oder `POWERING_DOWN` state und jeder Knoten, der nicht heruntergefahren ist, meldet Version B.

Option 2: Full-fleet Wartungsstopp

Sie beenden die gesamte Flotte, bevor Sie den Controller aktualisieren, und skalieren ihn dann von einem neuen AMI aus mit der Ziel-Slurm-Version wieder hoch. Dieses Verfahren ist einfacher, aber es beendet alle Knoten und laufenden Jobs.

Wann sollte man Folgendes verwenden:

- Der Cluster-Controller ist auf Version 23.11 (Option 1 ist für 23.11-Cluster nicht verfügbar).

> **Note**
>
> Wenn die gesamte Flotte auf einmal beendet wird, erhöht sich die Wahrscheinlichkeit, dass bei der erneuten Skalierung Fehler aufgrund unzureichender Kapazität auftreten. Erwägen Sie die Nutzung reservierter Kapazitäten oder die Planung außerhalb der Spitzenzeiten.

Schritt 0 — Überprüfen Sie den Startstatus

Auf Ihrem Cluster wird Controller-Version „A“ (z. B. 24.11) ausgeführt, und Sie möchten auf Version „B“ (z. B. 25.11) migrieren. Vergewissern Sie sich, dass alle Rechenknoten in Ihrer Flotte dieselbe Hauptversion ausführen, indem Sie diesen Befehl von einem Cluster-Knoten aus verwenden:

```
sinfo -N -o "%N %T %v"

# Example output:
#   NODELIST    STATE VERSION
#   compute-1   idle  24.11.7
#   compute-2   idle  24.11.7
```

Bestätigen Sie AWS PCS-Agent-Version auf einem Rechenknoten. Stellen Sie mit Systems Manager eine Verbindung zum Knoten her und überprüfen Sie das Bootstrap-Protokoll:

```
grep "PCS Agent version" /var/log/amazon/pcs/bootstrap.log | tail -1

# Example output:
#   /opt/aws/pcs/bin/pcs_bootstrap_init.sh: INFO: Bootstrap starting with PCS Agent
 version: 1.3.2-1
```

Verwenden Sie die neueste Version AWS PCS-Agent auf Ihren Ziel-AMIs. Weitere Informationen finden Sie unter AWS PCS-Agent-Versionen.

Schritt 1 — Bereiten Sie die Ziel-AMIs vor

Erstellen oder identifizieren Sie AMIs, die Folgendes beinhalten Slurm Version B und die neueste Version AWS PCS-Agent.

- Sie können das neueste verwenden PCS-ready DLA Mis. Solche AMIs werden mit den letzten drei unterstützten Slurm-Versionen ausgeliefert. Weitere Informationen finden Sie unter Verwenden von PCS-ready DLAMI mit AWS 5 STCK.

- Sie können eine bauen benutzerdefiniertes AMI, indem Sie den Installationsschritten für Slurm-Pakete folgen und AWS PCS-Agent. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.
- Wir empfehlen das nicht AWS PCS-Beispiel-AMI für den Produktionsgebrauch. Diese AMIs dienen nur zum Testen.

Schritt 2 — Verkleinern Sie die gesamte Flotte

Nehmen Sie den Strom auf `minNodeCount` und `maxNodeCount` für jede Rechenknotengruppe — diese werden Sie in Schritt 4 wiederherstellen.

```
for cng in $(aws pcs list-compute-node-groups --cluster-identifier cluster-id --query
 "computeNodeGroups[].id" --output text); do
    aws pcs get-compute-node-group \
      --cluster-identifier cluster-id \
      --compute-node-group-identifier "$cng" \
      --query "computeNodeGroup.
{Id:id,AmiId:amiId,Min:scalingConfiguration.minInstanceCount,Max:scalingConfiguration.maxInstan
 \
      --output table
  done
```

> **Warning**
>
> Der folgende Vorgang beendet alle laufenden Knoten und die darauf befindlichen Jobs.

Stellen Sie ein `minNodeCount` und `maxNodeCount` zu `0` auf jeder Rechenknotengruppe:

```
aws pcs update-compute-node-group \
  --cluster-identifier cluster-id \
  --compute-node-group-identifier cng-id \
  --scaling-configuration '{"minNodeCount": 0, "maxNodeCount": 0}'
```

Stellen Sie sicher, dass keine Instanzen markiert sind `aws:pcs:cluster-id` Ihr Cluster wird entsprechend ausgeführt, bevor Sie fortfahren:

```
aws ec2 describe-instances \
```

```
    --filters "Name=tag:aws:pcs:cluster-id,Values=cluster-id" \
    --query "Reservations[].Instances[].[InstanceId,ImageId,State.Name]" \
    --output table
```

Schritt 3 — Aktualisieren Sie den Cluster-Controller

AWS-Managementkonsole

1. Öffnen Sie den AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, der aktualisiert werden soll, und wählen Sie Bearbeiten.
4. Unter Einzelheiten zum Cluster, wählen Sie die Ziel-Scheduler-Version aus dem Scheduler Dropdown.
5. Wählen Aktualisieren um das Versionsupdate einzureichen.
6. Überwachen Sie den Cluster-Status. Der Cluster wird angezeigt als `UPDATING` während des Updates und kehrt zurück zu `ACTIVE` wenn abgeschlossen. Das Update ist in der Regel in 5—15 Minuten abgeschlossen.

AWS CLI

```
aws pcs update-cluster \
  --cluster-identifier cluster-id \
  --scheduler version=25.11
```

Warten Sie, bis der Cluster zu zurückgekehrt ist `ACTIVE`. Das Update ist in der Regel in 5—15 Minuten abgeschlossen.

Wenn der Cluster nicht zurückkehrt zu `ACTIVE` oder `UPDATE_FAILED` innerhalb von 30 Minuten, Kontakt AWS Unterstützung für Unterstützung.

Schritt 4 — Compute-Knotengruppen aktualisieren und Kapazität wiederherstellen

Stellen Sie für jede Rechenknotengruppe das neue AMI ein und stellen Sie die ursprünglichen Mindest- und Höchstkapazitätsgrenzen wieder her:

```
aws pcs update-compute-node-group \
  --cluster-identifier cluster-id \
```

```
  --compute-node-group-identifier cng-id \
  --ami-id new-ami-id \
  --scaling-configuration '{"minNodeCount": previous-min, "maxNodeCount": previous-
max}'
```

Der Cluster wird wieder hochskaliert. Auf allen neuen Knoten wird Slurm Version B mit der neuesten Version ausgeführt AWS PCS-Agent.

Beispiel: Aktualisierung über mehrere Versionen hinweg

Wenn sich die Zielversion außerhalb des Kompatibilitätsfensters Ihrer aktuellen Version befindet, müssen Sie den Controller durch eine oder mehrere Zwischenversionen verschieben und ihn Schritt für Schritt aktualisieren. Jeder Hop muss auf eine unterstützte Version innerhalb des Kompatibilitätsfensters der aktuellen Controller-Version abzielen.

Weil Option 2: Full-fleet Wartungsstopp skaliert die Flotte auf Null, bevor der Controller aktualisiert wird. Während der Controller zwischen den Versionen wechselt, laufen keine Rechenknoten. Daher können Ihre AMIs die verwenden endgültig Zielversion direkt — nur das Controller-Update (Schritt 3) wird für jeden Hop wiederholt.

Das folgende Beispiel aktualisiert einen Cluster von 23.11 auf 25.11 mithilfe des Option-2-Verfahrens. 23.11 liegt außerhalb des Kompatibilitätsfensters von 25.11, sodass der Controller in zwei Schritten aktualisiert wird (23.11 auf 25.05, dann 25.05 auf 25.11). Folgen Sie den Schritten von Option 2, wobei Schritt 3 in ein Update pro Hop aufgeteilt ist:

1. Schritt 1 — Bereiten Sie die Ziel-AMIs vor. Erstellen oder identifizieren Sie AMIs mit dem endgültig Version (25.11) und die neueste AWS PCS-Agent. Siehe Schritt 1 — Bereiten Sie die Ziel-AMIs vor.
2. Schritt 2 — Verkleinern Sie die gesamte Flotte. Notieren Sie die aktuelle Kapazität (siehe Schritt 2 — Verkleinern Sie die gesamte Flotte), und setzen Sie dann jede Rechenknotengruppe auf Null.

   ```
   aws pcs update-compute-node-group \
   --cluster-identifier my-cluster \
   --compute-node-group-identifier my-cng \
   --scaling-configuration '{"minNodeCount": 0, "maxNodeCount": 0}'
   ```

3. Schritt 3a — Aktualisieren Sie den Controller von 23.11 auf 25.05. Warten Sie, bis der Cluster zu zurückgekehrt ist `ACTIVE`.

   ```
   aws pcs update-cluster --cluster-identifier my-cluster \
   ```

```
--scheduler version=25.05
```

4. Schritt 3b — Aktualisieren Sie den Controller von 25.05 auf 25.11. Warten Sie, bis der Cluster zu zurückgekehrt ist `ACTIVE`.

```
aws pcs update-cluster --cluster-identifier my-cluster \
--scheduler version=25.11
```

5. Schritt 4 — Aktualisieren Sie die Rechenknotengruppen und stellen Sie die Kapazität wieder her. Stellen Sie das 25.11-AMI für jede Rechenknotengruppe ein und stellen Sie die ursprünglichen Kapazitätsgrenzen wieder her (siehe Schritt 4 — Compute-Knotengruppen aktualisieren und Kapazität wiederherstellen).

```
aws pcs update-compute-node-group \
--cluster-identifier my-cluster \
--compute-node-group-identifier my-cng \
--ami-id ami-0123456789abcdef0 \
--scaling-configuration '{"minNodeCount": previous-min, "maxNodeCount": previous-
max}'
```

> **Note**
>
> Jeder Controller-Hop muss auf einer Version innerhalb des Kompatibilitätsfensters der vorherigen Version landen. Informationen zur Suche nach gültigen Zwischenversionen finden Sie unter Versionskompatibilität. Die Flotte bleibt während der Schritte 3a und 3b auf Null, sodass keine zwischengeschalteten AMI-Aktualisierungen erforderlich sind.

Fehlerbehebung AWS Aktualisierungen der PCS-Cluster-Versionen

Dieses Thema hilft Ihnen dabei, häufig auftretende Probleme zu identifizieren und zu lösen, die bei der Aktualisierung der Scheduler-Version auf einem Cluster auftreten können.

- Rechenknoten können nach dem Update keine Verbindung herstellen
- Aktualisierungsanfrage abgelehnt mit ValidationException
- Der Cluster befindet sich weiterhin im Status UPDATE
- Nach dem Update wechselt der Cluster in den Status UPDATE_FAILED
- Rechenknoten können aufgrund von Fehlern bei der Konfigurationsanalyse nicht gestartet werden

- Cluster nach dem Update aufgrund einer ungültigen QoS-Konfiguration nicht verfügbar

Rechenknoten können nach dem Update keine Verbindung herstellen

Häufige Ursache

Nach der Aktualisierung des Clusters können sich neu gestartete Rechenknoten nicht beim Controller registrieren. Die Knoten werden gestartet, erscheinen aber nie in der `sinfo` Ausgabe, und die Compute-Knotengruppe durchläuft möglicherweise wiederholt Instanzen. Dies tritt auf, wenn das AMI der Compute-Knotengruppe eine Scheduler-Version enthält, die außerhalb des Kompatibilitätsfensters der neuen Controller-Version liegt.

Wenn Sie beispielsweise einen Cluster von 24.11 auf 25.11 aktualisieren, die Compute-Knotengruppe aber immer noch ein AMI mit Slurm 23.11 verwendet, können neue Instances keine Verbindung herstellen, da 23.11 außerhalb des Kompatibilitätsfensters von 25.11 liegt.

Wie diagnostiziert man

Sie können dieses Problem anhand der Logs an zwei Stellen überprüfen:

Scheduler-Protokolle

Wenn Sie die Scheduler-Protokollierung aktiviert haben, überprüfen Sie die Scheduler-Protokolle unter Logs auf CloudWatch Fehler, die den folgenden ähneln:

```
error: unpack_header: protocol_version 10240 not supported
error: slurm_unpack_received_msg: [ip-10-0-1-23] Incompatible versions of client and
 server code
```

Diese Fehler deuten darauf hin, dass ein Rechenknoten mit einer inkompatiblen Scheduler-Version versucht, eine Verbindung zum Controller herzustellen. Hinweise zum Einrichten der Scheduler-Protokollierung finden Sie unter. Scheduler loggt sich in AWS PCS ein

Logs für Knoteninstanzen berechnen

Rufen Sie die Ausgabe der Instanzkonsole ab oder stellen Sie eine Verbindung über Systems Manager her und überprüfen Sie das Bootstrap-Protokoll auf Fehler, die den folgenden ähneln:

```
error: _fetch_child: failed to fetch remote configs: Incompatible versions of client
 and server
error: _establish_configuration: failed to load configs
```

```
error: slurmd initialization failed
```

Weitere Informationen zum Abrufen von Instanzprotokollen finden Sie unter. Rufen Sie Instanzprotokolle ab

Auflösung

Aktualisieren Sie die Compute-Knotengruppe, um ein AMI zu verwenden, das eine Scheduler-Version im Kompatibilitätsfenster Ihrer neuen Cluster-Version enthält:

```
aws pcs update-compute-node-group \
--cluster-identifier my-cluster \
--compute-node-group-identifier my-cng \
--ami-id ami-0123456789abcdef0
```

Informationen darüber, welche Scheduler-Versionen mit Ihrem Cluster kompatibel sind, finden Sie unter. Versionskompatibilität

Informationen zum Erstellen benutzerdefinierter AMIs mit der richtigen Scheduler-Version finden Sie unter. Amazon Machine Images (AMIs) für AWS STK.

Aktualisierungsanfrage abgelehnt mit ValidationException

Häufige Ursache

Die `UpdateCluster` Anfrage kehrt sofort mit einem `ValidationException` Fehler zurück, der darauf hinweist, dass das Update nicht unterstützt wird. Dies tritt auf, wenn:

- Die Zielversion befindet sich außerhalb des Kompatibilitätsfensters der aktuellen Version.
- Die Zielversion wird als End of Life (EOL) bezeichnet und ist kein gültiges Aktualisierungsziel mehr.
- Die Zielversion ist älter als die aktuelle Version oder entspricht ihr (Downgrades werden nicht unterstützt).

Auflösung

Wenn sich die Zielversion außerhalb des Kompatibilitätsfensters befindet, führen Sie das Update in mehreren Schritten durch. Jeder Schritt muss auf eine unterstützte Version innerhalb des Kompatibilitätsfensters abzielen. Um beispielsweise von 23.11 auf 25.11 zu wechseln, aktualisieren Sie zuerst auf 25.05, warten Sie, bis der Cluster wieder auf 25.11 zurückkehrt, und aktualisieren Sie dann auf `ACTIVE` 25.11.

Wenn die Zielversion EOL ist, wählen Sie stattdessen eine neuere unterstützte Version. Informationen zu unterstützten Versionen finden Sie unterSlurm-Versionen in AWS 5 STCK.

Der Cluster befindet sich weiterhin im Status UPDATE

Häufige Ursache

Der Cluster bleibt länger als erwartet im `UPDATING` Status (mehr als 20 Minuten). Dies kann aufgrund vorübergehender interner Probleme während des Aktualisierungsvorgangs auftreten.

Auflösung

AWS PCS stellt automatisch Cluster wieder her, die im `UPDATING` Status feststecken. Wenn der Cluster nicht zu `ACTIVE` oder `UPDATE_FAILED` innerhalb von 30 Minuten zurückkehrt, wenden Sie sich an den AWS Support, um Unterstützung zu erhalten.

Nach dem Update wechselt der Cluster in den Status UPDATE_FAILED

Häufige Ursache

Der Cluster wechselt während des Updates in den `UPDATE_FAILED` Status. Dies kann der Fall sein, wenn vorübergehende Dienstfehler verhindern, dass das Update erfolgreich abgeschlossen wird.

Auflösung

Versuchen Sie das Update erneut, indem Sie dieselbe `UpdateCluster` Anfrage erneut senden. Cluster im `UPDATE_FAILED` Status akzeptieren neue Aktualisierungsanfragen. Wenn das Update weiterhin fehlschlägt, wenden Sie sich an den AWS Support.

Rechenknoten können aufgrund von Fehlern bei der Konfigurationsanalyse nicht gestartet werden

Häufige Ursache

Nach der Aktualisierung des Clusters können die Rechenknoten nicht gestartet werden und erscheinen nie in der `sinfo` Ausgabe. In den Protokollen der Compute-Knoteninstanz werden Fehler angezeigt, die den folgenden ähneln:

```
error: _parse_next_key: Parsing error at unrecognized key: HashPlugin
error: Invalid DebugFlag: AuditRPCs
fatal: Unable to process configuration file
```

Dies tritt auf, wenn Compute-Knoten, auf denen die Scheduler-Version 23.11 ausgeführt wird, eine Konfiguration von einer neueren Cluster-Version erhalten. In Scheduler-Versionen nach 23.11 wurden neue Konfigurationsdirektiven eingeführt, die 23.11 nicht analysieren kann. Im Gegensatz zu anderen Versionen innerhalb des Kompatibilitätsfensters können 23.11-Compute-Knoten keine Verbindung zu einem neueren Cluster herstellen, da sie bei unbekannten Konfigurationsschlüsseln fatale Fehler machen.

Dieses Problem kann auch auftreten, wenn Ihr benutzerdefiniertes AMI eine AWS PCS-Agent-Version verwendet, die älter als v1.4.0 ist. Ältere Agentenversionen unterstützen keinen automatischen Versions-Fallback für den Compute Node Daemon.

Auflösung

Erstellen Sie Ihr benutzerdefiniertes AMI mit den folgenden Anforderungen neu:

- Scheduler Version 24.05 oder höher
- AWS PCS Agent Version 1.4.0 oder höher

Aktualisieren Sie dann die Compute-Knotengruppe, um das neue AMI zu verwenden:

```
aws pcs update-compute-node-group \
--cluster-identifier my-cluster \
--compute-node-group-identifier my-cng \
--ami-id ami-0123456789abcdef0
```

Informationen zum Erstellen benutzerdefinierter AMIs finden Sie unterAmazon Machine Images (AMIs) für AWS STK.. Informationen zu den Versionen von AWS PCS-Agenten finden Sie unterAWS PCS-Agent-Versionen.

Cluster nach dem Update aufgrund einer ungültigen QoS-Konfiguration nicht verfügbar

Häufige Ursache

Nach dem Update auf Version 25.11 wechselt der Cluster in den `UPDATE_FAILED` Status oder der Scheduler ist nicht mehr verfügbar. Sie können keine Jobs einreichen oder Scheduler-Befehle ausführen. Die Scheduler-Protokolle zeigen Fehler, die den folgenden ähneln:

```
error: Invalid Allow/DenyQOS value: low
fatal: Partition my-queue has an invalid DenyQOS (low), please check your configuration
```

Dies tritt auf, wenn eine Warteschlange (Partition) über`AllowQOS`,, oder `QOS` Einstellungen auf einen QoS-Namen verweist`DenyQOS`, die in der Slurm-Accounting-Datenbank nicht vorhanden sind. Slurm 25.11 führte eine strengere Validierung von QoS-Referenzen beim Start des Schedulers ein. Frühere Versionen erlaubten Verweise auf nicht existierende QoS-Namen ohne Fehler.

Auflösung

Stellen Sie vor dem Update auf Version 25.11 sicher, dass alle QoS-Namen, auf die in Ihren Warteschlangenkonfigurationen verwiesen wird, in der Accounting-Datenbank vorhanden sind. Stellen Sie eine Connect zu einem Anmeldeknoten her und führen Sie den folgenden Befehl aus, um zu überprüfen, ob eine QoS vorhanden ist:

```
sacctmgr show qos where name=low format=name
```

Wenn die QoS nicht existiert, erstellen Sie sie, bevor Sie das Update versuchen:

```
sacctmgr add qos low
```

Alternativ können Sie die QoS-Referenz aus Ihrer Warteschlangenkonfiguration entfernen, indem Sie die benutzerdefinierten Slurm-Einstellungen der Warteschlange aktualisieren, um den `QOS` Parameter`AllowQOS`, oder zu entfernen`DenyQOS`, bevor Sie den Cluster aktualisieren.

Häufig gestellte Fragen zum Aktualisieren von Clustern in AWS STCK

Hier finden Sie Antworten auf häufig gestellte Fragen zur Aktualisierung von Clusterkonfigurationen in AWS PCS.

Welche Einstellungen kann ich ändern?

> Sie können die Accounting-Konfiguration (enable/disable Accounting), das Scale-Down-Verhalten (DownIdleTime Skalierungsparameter), die Scheduler-Version und alle unterstützten benutzerdefinierten Slurm-Einstellungen, die auf Cluster-Ebene gelten, ändern. Sie können Sicherheitsgruppen, VPC-Subnetze, Clustergröße oder Clusternamen nicht ändern.

Kann ich mehrere Updates in die Warteschlange stellen?

> Nein. Sie müssen warten, bis der Cluster wieder in den `ACTIVE` Status zurückkehrt, bevor Sie ein weiteres Update einreichen. Alle zugehörigen Ressourcen (Warteschlangen, Compute Node Groups) müssen ebenfalls im `ACTIVE` Status sein.

Kann ich einen Cluster-Aktualisierungsvorgang abbrechen?

Nein, Sie können einen laufenden Cluster-Aktualisierungsvorgang nicht abbrechen.

Kann ich Jobs einreichen, während mein Cluster aktualisiert wird?

Wir empfehlen Ihnen, das Senden von Aufträgen während der Cluster-Updates zu vermeiden. Der Slurm-Controller ist während des Aktualisierungsvorgangs möglicherweise nicht verfügbar.

Laufen meine Jobs während der Cluster-Updates weiter?

Ja, laufende Jobs werden weiterhin auf Rechenknoten ausgeführt, auch wenn der Cluster-Controller während des Aktualisierungsvorgangs kurzzeitig nicht mehr erreichbar ist. Der Auftragsstatus wird jedoch möglicherweise erst aktualisiert, wenn der Controller wieder verfügbar ist.

Wie aktualisiere ich eigenständige Anmeldeknoten während eines Scheduler-Versionsupdates?

Eigenständige Anmeldeknoten sind EC2-Instances, die nicht von AWS PCS-Rechenknotengruppen verwaltet werden. AWS PCS aktualisiert diese Instances nicht automatisch während eines Scheduler-Versionsupdates. Um sie zu aktualisieren, ersetzen Sie sie durch neue Instances, die ein AMI verwenden, das die Ziel-Slurm-Version und den neuesten AWS PCS-Agenten enthält. Der Zeitpunkt hängt davon ab, welche Aktualisierungsoption Sie verwenden:

- Option 1 (fortlaufendes Update) — Aktualisieren Sie die Anmeldeknoten, nachdem das Controller-Update abgeschlossen ist und die Flotte vollständig auf der neuen Version läuft.
- Option 2 (vollständige Einstellung der Wartung der Flotte) — Aktualisieren Sie die Anmeldeknoten zu einem beliebigen Zeitpunkt während des Vorgangs.

Weitere Informationen finden Sie unter Aktualisieren Sie die Scheduler-Version eines AWS PCS-Cluster.

Wie wirkt sich die Abrechnung bei Updates aus?

Während des Aktualisierungsvorgangs werden weiterhin die üblichen stündlichen Gebühren berechnet. Wenn das Accounting deaktiviert wird, stoppt die Abrechnung, sobald der Cluster in den `UPDATING` Status übergeht. Wenn Sie Accounting aktivieren, beginnt die Abrechnung, wenn der Cluster erfolgreich in den `ACTIVE` Status zurückkehrt.

Problembehandlung bei AWS PCS-Cluster-Updates

Dieses Thema hilft Ihnen dabei, häufig auftretende Probleme zu identifizieren und zu lösen, die bei der Aktualisierung von Clusterkonfigurationen auftreten können.

Das Update schlägt mit einem Fehler bei der Kontoführungskonfiguration fehl

Häufige Ursache

Der Cluster wechselt in den `UPDATE_FAILED` Status und die Fehlermeldung weist auf ein Problem mit der Kontoführungskonfiguration hin. Dies tritt normalerweise auf, wenn die Accounting-Konfiguration nicht mit der aktuellen Slurm-Version kompatibel ist oder ungültige Einstellungen enthält.

Auflösung

Überprüfen Sie Ihre Accounting-Einstellungen auf Kompatibilität mit der Slurm-Version Ihres Clusters und reichen Sie eine korrigierte Aktualisierungsanfrage mit gültigen Konfigurationsparametern ein.

Das Update schlägt mit einem Fehler bei den benutzerdefinierten Einstellungen fehl

Häufige Ursache

Der Cluster wechselt in den `UPDATE_FAILED` Status und die Fehlermeldung weist auf ein Problem mit den benutzerdefinierten Slurm-Einstellungen hin. Dies tritt auf, wenn Sie ungültige Slurm-Parameterwerte oder nicht unterstützte Parameterkombinationen angeben.

Auflösung

Überprüfen Sie Ihre benutzerdefinierten Slurm-Einstellungen anhand der unterstützten Parameter und senden Sie eine korrigierte Aktualisierungsanforderung mit gültigen Parameterwerten und Kombinationen.

Aktualisierungsanfrage kann nicht eingereicht werden

Häufige Ursache

Die Aktualisierungsschaltfläche ist in der Konsole deaktiviert oder die API gibt einen Fehler der Stufe 400 zurück. Dies tritt auf, wenn sich der Cluster nicht in einem geeigneten Zustand befindet, die zugehörigen Ressourcen nicht aktiv sind oder wenn in Ihrer Konfiguration Validierungsfehler vorliegen.

Auflösung

Warten Sie, bis der Cluster und alle zugehörigen Ressourcen den `ACTIVE` Status erreicht haben, und überprüfen Sie dann Ihre Konfiguration auf Validierungsfehler, bevor Sie die Aktualisierungsanforderung erneut einreichen.

Validierungsfehler

Häufige Ursache

Der Befehl kehrt sofort mit einem HTTP-Fehler der Stufe 400 und einer beschreibenden Meldung zurück. Dies ist auf ungültige Clusterstatus-, Ressourcenstatus- oder Konfigurationsparameter zurückzuführen.

Auflösung

Beheben Sie den spezifischen Validierungsfehler, der in der Antwort erwähnt wurde, und wiederholen Sie den Aktualisierungsvorgang.

Löschen eines Clusters in AWS PCS

Dieses Thema bietet einen Überblick darüber, wie Sie einen AWS-PCS-Cluster löschen.

Überlegungen beim Löschen eines AWS PCS-Clusters

- Alle mit dem Cluster verknüpften Warteschlangen müssen gelöscht werden, bevor der Cluster gelöscht werden kann. Weitere Informationen finden Sie unter Löschen einer Warteschlange in AWS PCS.
- Alle mit dem Cluster verknüpften Compute-Knotengruppen müssen gelöscht werden, bevor der Cluster gelöscht werden kann. Weitere Informationen finden Sie unter Löschen einer Compute-Knotengruppe in AWS PCS.

Löschen Sie den Cluster

Sie können das AWS-Managementkonsole oder verwenden AWS CLI , um einen Cluster zu löschen.

AWS-Managementkonsole

So löschen Sie einen Cluster

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den zu löschenden Cluster aus.
3. Wählen Sie Löschen aus.
4. Das Feld Cluster-Status wird angezeigt`Deleting`. Das kann mehrere Minuten dauern.

AWS CLI

So löschen Sie einen Cluster

1. Verwenden Sie den folgenden Befehl, um einen Cluster mit diesen Ersetzungen zu löschen:

   - Ersetzen Sie *region-code* durch den, in dem sich AWS-Region Ihr Cluster befindet.
   - *my-cluster*Ersetzen Sie durch den Namen oder die ID Ihres Clusters.

   ```
   aws pcs delete-cluster --region region-code --cluster-identifier my-cluster
   ```

2. Das Löschen des Clusters kann mehrere Minuten dauern. Sie können den Status Ihres Clusters mit dem folgenden Befehl überprüfen.

   ```
   aws pcs get-cluster --region region-code --cluster-identifier my-cluster
   ```

Clustergröße in AWS PCS

AWS PCS bietet hochverfügbare und sichere Cluster und automatisiert gleichzeitig wichtige Aufgaben wie Patching, Knotenbereitstellung und Updates.

Wenn Sie einen Cluster erstellen, wählen Sie dessen Größe auf der Grundlage von zwei Faktoren aus:

- Die Anzahl der Rechenknoten, die verwaltet werden
- Die Anzahl der Jobs, die vom Controller zu einem bestimmten Zeitpunkt verfolgt wurden

> **Note**
>
> Die Anzahl der Jobs umfasst laufende, ausstehende und kürzlich abgeschlossene Jobs. Abgeschlossene Jobs werden vom Controller für einen kurzen Zeitraum nachverfolgt, bevor sie gelöscht werden. Bei hohem Auftragsdurchsatz kann dies dazu führen, dass die Gesamtzahl der verfolgten Jobs die Anzahl der aktiven Jobs, die Sie beobachten, übersteigt.

> **Important**
>
> Sie können die Clustergröße nicht ändern, nachdem Sie den Cluster erstellt haben. Wenn Sie die Größe ändern müssen, müssen Sie einen neuen Cluster erstellen.

Größe des Slurm-Clusters	Anzahl der verwalteten Instanzen	Anzahl der vom Controller verfolgten Jobs
Small	Bis zu 32	Bis zu 256
Mittel	Bis zu 512	Bis zu 8192
Large (Groß)	Bis zu 2048	Bis zu 16384

Beispiele

- Wenn Ihr Cluster über bis zu 24 verwaltete Instanzen verfügen und bis zu 100 Jobs ausführen soll, wählen Sie Small.
- Wenn Ihr Cluster über bis zu 24 verwaltete Instanzen verfügen und bis zu 1000 Jobs ausführen soll, wählen Sie Medium.
- Wenn Ihr Cluster bis zu 1000 verwaltete Instanzen haben und bis zu 100 Jobs ausführen soll, wählen Sie Large.
- Wenn Ihr Cluster bis zu 1000 verwaltete Instanzen haben und bis zu 10.000 Jobs ausführen soll, wählen Sie Large.

Arbeiten mit Cluster-Secrets in AWS STCK

Im Rahmen der Clustererstellung AWS PCS erstellt ein Cluster-Geheimnis, das für die Verbindung zum Job-Scheduler auf dem Cluster erforderlich ist. Sie erstellen auch AWS PCS-Rechenknotengruppen, die Gruppen von Instanzen definieren, die als Reaktion auf Skalierungsereignisse gestartet werden. AWS PCS konfiguriert Instances, die von diesen Compute-Knotengruppen gestartet werden, mit dem Cluster-Schlüssel, sodass sie eine Verbindung zum Job Scheduler herstellen können. Es gibt Fälle, in denen Sie Slurm-Clients möglicherweise manuell konfigurieren möchten. Beispiele hierfür sind der Aufbau eines persistenten Anmeldeknotens oder die Einrichtung eines Workflow-Managers mit Funktionen zur Auftragsverwaltung.

AWS PCS speichert das Cluster-Geheimnis als verwaltetes Geheimnis mit dem Präfix `pcs!` in AWS Secrets Manager. Die Kosten für das Geheimnis sind in der Nutzungsgebühr enthalten AWS 5 STCK. Sie können Cluster-Secrets rotieren durch AWS Secrets Manager um die Einhaltung der Sicherheitsvorschriften zu gewährleisten und potenzielle Sicherheitslücken zu beheben.

Themen

- Verwenden Sie AWS Secrets Manager um das Cluster-Geheimnis zu finden
- Verwenden Sie AWS PCS, um das Cluster-Geheimnis zu finden
- Hol dir das Geheimnis des Slurm-Clusters
- Verwenden Sie einen vom Kunden verwalteten Schlüssel, um das Cluster-Geheimnis zu verschlüsseln
- Rotierende Clustergeheimnisse in AWS PCS

Verwenden Sie AWS Secrets Manager um das Cluster-Geheimnis zu finden

AWS-Managementkonsole

1. Navigieren Sie zum Secrets Manager-Konsole.
2. Wählen Geheimnisse, dann suche nach dem `pcs!` Präfix.

> **Note**
>
> EIN AWS Das PCS-Clustergeheimnis hat einen Namen in der Form `pcs!slurm-secret-`*`cluster-id`* woher *`cluster-id`* ist der AWS PCS-Cluster-ID.

AWS CLI

Jeder AWS PCS Cluster Secret ist auch markiert mit `aws:pcs:cluster-id`. Sie können die geheime ID für einen Cluster mit dem folgenden Befehl abrufen. Nehmen Sie diese Ersetzungen vor, bevor Sie den Befehl ausführen:

- Ersetzen *`region`* mit dem AWS-Region um Ihren Cluster in zu erstellen, wie `us-east-1`.
- Ersetzen *`cluster-id`* mit der ID des AWS PCS-Cluster, für den das Cluster-Geheimnis gefunden werden soll.

```
aws secretsmanager list-secrets \
    --region region  \
    --filters Key=tag-key,Values=aws:pcs:cluster-id \
            Key=tag-value,Values=cluster-id
```

Verwenden Sie AWS PCS, um das Cluster-Geheimnis zu finden

Sie können das verwenden AWS CLI um den ARN für einen zu finden AWS PCS-Cluster-Geheimnis. Geben Sie den folgenden Befehl ein und nehmen Sie die folgenden Ersetzungen vor:

- Ersetzen *region* mit dem AWS-Region um Ihren Cluster in zu erstellen, wie `us-east-1`.
- Ersetzen *my-cluster* mit dem Namen oder der Kennung für Ihren Cluster.

```
aws pcs get-cluster --region region --cluster-identifier my-cluster
```

Die folgende Beispielausgabe stammt von `get-cluster` Befehl. Du kannst verwenden `secretArn` und `secretVersion` zusammen, um das Geheimnis zu lüften.

```
{
    "cluster": {
        "name": "get-started",
        "id": "pcs_123456abcd",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
        "status": "ACTIVE",
        "createdAt": "2024-12-17T21:03:52+00:00",
        "modifiedAt": "2024-12-17T21:03:52+00:00",
        "scheduler": {
            "type": "SLURM",
            "version": "25.11"
        },
        "size": "SMALL",
        "slurmConfiguration": {
            "authKey": {
                "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!
slurm-secret-pcs_123456abcd-a12ABC",
                "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
            }
        },
        "networking": {
```

```
            "subnetIds": [
                "subnet-0123456789abcdef0"
            ],
            "securityGroupIds": [
                "sg-0123456789abcdef0"
            ]
        },
        "endpoints": [
            {
                "type": "SLURMCTLD",
                "privateIpAddress": "10.3.149.220",
                "port": "6817"
            }
        ]
    }
}
```

Hol dir das Geheimnis des Slurm-Clusters

Sie können Secrets Manager verwenden, um die aktuelle Version von abzurufen Slurm geheimes Cluster-Geheimnis. Secrets Manager gibt das Geheimnis als Base64-kodierte Zeichenfolge zurück. Das folgende Beispiel verwendet die AWS CLI. Nehmen Sie die folgenden Ersetzungen vor, bevor Sie den Befehl ausführen.

- Ersetzen *`region`* mit dem AWS-Region um Ihren Cluster in zu erstellen, wie `us-east-1`.
- Ersetzen *`secret-arn`* mit dem `secretArn` von einem AWS PCS-Cluster.

```
aws secretsmanager get-secret-value \
    --region region \
    --secret-id 'secret-arn' \
    --version-stage AWSCURRENT \
    --query 'SecretString' \
    --output text
```

> **Important**
>
> Secrets Manager speichert und gibt das Cluster-Geheimnis als Base64-kodierte Zeichenfolge zurück, aber Slurm benötigt das Geheimnis als rohe (dekodierte) Byte. Wenn Sie das Geheimnis in eine schreiben Slurm Schlüsseldatei (zum Beispiel `/etc/slurm/slurm.key`), müssen Sie den Wert zuerst mit Base64 dekodieren. Wenn Sie die Base64-kodierte

Zeichenfolge direkt schreiben, wird eine Schlüsseldatei erzeugt, die Slurm kann nicht verwendet werden. `slurmd` schlägt dann die Authentifizierung mit einem Fehler wie folgt fehl `decode_jwt: jwt_decode (with key) failure: Invalid argument`.
Um das Geheimnis zu dekodieren und in eine Schlüsseldatei zu schreiben, leiten Sie die Ausgabe durch `base64 -d`:

```
aws secretsmanager get-secret-value \
    --region region \
    --secret-id 'secret-arn' \
    --version-stage AWSCURRENT \
    --query 'SecretString' \
    --output text | base64 -d | sudo tee /etc/slurm/slurm.key
```

> **Note**
>
> Sie müssen das Cluster-Geheimnis nicht selbst auf einem abrufen AWS PCS-managed Knotengruppe berechnen. AWS PCS übermittelt das Geheimnis an Instanzen, die von der Compute-Knotengruppe als Teil des Node-Bootstraps gestartet wurden. Rufen Sie das Geheimnis nur manuell ab, wenn Sie eine konfigurieren Slurm Client außerhalb einer verwalteten Rechenknotengruppe, z. B. ein persistenter Anmeldeknoten.

Hinweise zur Verwendung des Slurm-Cluster-Geheimnisses finden Sie unter Verwenden eigenständiger Instanzen als AWS PCS-Anmeldeknoten.

Berechtigungen

Sie verwenden einen IAM-Principal, um das Geheimnis des Slurm-Clusters zu erhalten. Der IAM-Principal muss die Erlaubnis haben, das Geheimnis zu lesen. Weitere Informationen finden Sie unter Begriffe und Konzepte von Rollen in der AWS Identity and Access Management Benutzerleitfaden.

Die folgende Beispiel-IAM-Richtlinie ermöglicht den Zugriff auf ein Beispiel für einen Cluster-Schlüssel.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AllowSecretValueRetrievalAndVersionListing",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:ListSecretVersionIds"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:012345678901:secret:pcs!
slurm-secret-s3431v9rx2-FN7tJF"
        }
    ]
}
```

Verwenden Sie einen vom Kunden verwalteten Schlüssel, um das Cluster-Geheimnis zu verschlüsseln

Standardmäßig speichert AWS PCS das Cluster-Geheimnis AWS Secrets Manager verschlüsselt mit einem Von AWS verwalteter Schlüssel. AWS PCS benötigt keine zusätzliche Autorisierung, um den zu verwenden Von AWS verwalteter Schlüssel. Alternativ können Sie das Cluster-Geheimnis mit einem vom Kunden verwalteten Schlüssel verschlüsseln. Bei dem vom Kunden verwalteten Schlüssel muss es sich um einen symmetrischen Verschlüsselungsschlüssel (`SYMMETRIC_DEFAULT`,`ENCRYPT_DECRYPT`) im selben AWS-Konto Land AWS-Region wie der Cluster handeln.

Die AWS verwaltete Richtlinie`AWSPCSServiceRolePolicy`, die der dienstverknüpften AWS PCS-Rolle (`AWSServiceRoleForPCS`) zugeordnet ist, gewährt keine `kms:` Berechtigungen. Wenn Sie einen vom Kunden verwalteten Schlüssel an den geheimen Schlüssel anhängen, ohne die Schlüsselrichtlinie zu bearbeiten, kann die serviceverknüpfte Rolle nicht auf den geheimen Schlüssel zugreifen. Das hat zur Folge, dass geheime Rotations- und Rechenknotenoperationen fehlschlagen. Weitere Informationen zur dienstverknüpften AWS PCS-Rolle finden Sie unterService-linked Rollen für AWS 5 STÜCK.

Erforderliche wichtige Richtlinie

Fügen Sie der Schlüsselrichtlinie des vom Kunden verwalteten Schlüssels die folgende Aussage hinzu. Ersetze es *`account-id`* durch deine AWS-Konto ID. Diese Anweisung erteilt der vom AWS PCS Service verknüpften Rolle (`AWSServiceRoleForPCS`) die Erlaubnis, den vom Kunden verwalteten Schlüssel zum Schutz des Cluster-Geheimnisses zu verwenden.

```
{
   "Sid": "Allow service-linked role use of the customer managed key",
```

```
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Weitere Informationen finden Sie unter:

- https://docs.aws.amazon.com/cli/latest/reference/kms/put-key-policy.htmlput-key-policy in der Befehlsreferenz AWS CLI
- create-key in der Befehlsreferenz AWS CLI
- Die Schlüssel-ID und den Schlüssel-ARN finden Sie im Entwicklerhandbuch AWS Key Management Service
- AWS Key Management Serviceim AWS Key Management Service Entwicklerhandbuch
- Das AWS PCS-Cluster-Geheimnis wird als verwaltetes Geheimnis in gespeichert AWS Secrets Manager.

Berechtigungen zum Ändern des Verschlüsselungsschlüssels

Die Berechtigungen im vorherigen Abschnitt gelten für die dienstverknüpfte Rolle von AWS PCS. Der IAM-Prinzipal, der den an das Secret angehängten Verschlüsselungsschlüssel ändert, benötigt seine eigenen Berechtigungen, die von der dienstverknüpften Rolle getrennt sind. Der Principal benötigt die folgenden Berechtigungen:

- `secretsmanager:UpdateSecret`auf dem Cluster-Schlüssel.
- `kms:Decrypt`auf dem ausgehenden Schlüssel (der Schlüssel, der das Geheimnis derzeit verschlüsselt).
- `kms:GenerateDataKey``kms:Encrypt`, und `kms:Decrypt` auf dem eingehenden Schlüssel (dem neuen Schlüssel).

Wenn Sie den Schlüssel ändern, werden alle gespeicherten Versionen des Geheimnisses, einschließlich der Versionen `AWSCURRENT` und`AWSPREVIOUS`, AWS Secrets Manager erneut verschlüsselt.

Informationen zum Anhängen des Schlüssels finden Sie unter. Hängen Sie den Schlüssel an und drehen Sie das Geheimnis

Hängen Sie den Schlüssel an und drehen Sie das Geheimnis

Während des AWS Secrets Manager `UpdateSecret` Vorgangs hängen Sie den vom Kunden verwalteten Schlüssel an den geheimen Clusterschlüssel an. AWS PCS stellt keinen API-Parameter für den Verschlüsselungsschlüssel des Geheimnisses bereit.

Im folgenden Beispiel wird der vom Kunden verwaltete Schlüssel an das Cluster-Geheimnis angehängt. Ersetzen Sie die Platzhalter durch Ihre Werte.

```
aws secretsmanager update-secret \
    --region region \
    --secret-id secret-arn \
    --kms-key-id key-arn
```

> **Note**
>
> Bei dem vom Kunden verwalteten Schlüssel muss es sich um einen symmetrischen Verschlüsselungsschlüssel handeln, der mit dem geheimen Schlüssel identisch AWS-Konto ist. AWS-Region Sie benötigen die unter aufgeführten Berechtigungen. Berechtigungen zum Ändern des Verschlüsselungsschlüssels

Nachdem Sie den vom Kunden verwalteten Schlüssel angehängt haben, fahren Sie mit der Rotation fort, sodass AWS PCS einen neuen geheimen Clusterschlüssel generiert, der mit dem neu hinzugefügten kundenverwalteten Schlüssel verschlüsselt ist. Informationen zum Rotationsverfahren finden Sie unterRotiert ein Cluster-Geheimnis in AWS 5 STÜCK.

Rotierende Clustergeheimnisse in AWS PCS

Verwenden Sie AWS Secrets Manager Managed Rotation, um Cluster-Geheimnisse in AWS PCS rotieren zu lassen. Die regelmäßige Rotation von Geheimnissen ist eine bewährte Sicherheitsmethode zur Aufrechterhaltung eines hohen Sicherheitsniveaus in HPC-Umgebungen.

Diese Funktion ermöglicht es Ihnen, branchenübliche Compliance-Standards wie HIPAA und FedRAMP zu erfüllen, die eine regelmäßige Rotation von Anmeldeinformationen vorschreiben.

Das Clustergeheimnis dient zwei Zwecken: zur Authentifizierung von Rechenknoten, die dem Cluster beitreten, und als JWT-Schlüssel für die Slurm-REST-API-Authentifizierung. Bei Rotation wirken sich beide Aspekte gleichzeitig aus.

So funktioniert die geheime Cluster-Rotation

Bereiten Sie sich manuell vor, um die Cluster-Stabilität während der geheimen Rotation aufrechtzuerhalten:

1. Vorbereitung — Skalieren Sie alle Rechenknotengruppen auf eine Kapazität von 0 und stellen Sie sicher, dass keine Jobs ausgeführt werden
2. Rotation — Initiieren Sie die Rotation über die Secrets Manager Manager-Konsole oder API
3. Überwachung — Verfolgen Sie den Fortschritt anhand von CloudTrail Ereignissen
4. Wiederherstellung — Skalieren Sie die Rechenknotengruppen wieder auf die gewünschte Kapazität

Während der Rotation bleibt Ihr Cluster unverändert `ACTIVE` und die Abrechnung läuft normal weiter. Der Vorgang dauert in der Regel einige Minuten.

Anforderungen und Einschränkungen

Bevor Sie Clustergeheimnisse rotieren, müssen Sie die folgenden Anforderungen erfüllen:

- Der Cluster muss den `UPDATE_FAILED` Status „`ACTIVE`oder“ haben
- Die IAM-Rolle muss über eine entsprechende Berechtigung verfügen `secretsmanager:RotateSecret`
- Alle Compute-Knotengruppen müssen auf eine Kapazität von 0 skaliert werden
- Stoppen Sie alle Jobs vor der Rotation

Einschränkungen:

- Für jede Rotation ist eine manuelle Vorbereitung erforderlich
- Bestehende JWT-Token werden ungültig und müssen erneut ausgestellt werden

- BYO-Anmeldeknoten müssen nach der Rotation manuell geheim aktualisiert werden

Topics

- Rotiert ein Cluster-Geheimnis in AWS 5 STÜCK
- Häufig gestellte Fragen zur Cluster-Secret-Rotation in AWS STCK
- Fehlerbehebung bei der geheimen Cluster-Rotation in AWS PCS

Rotiert ein Cluster-Geheimnis in AWS 5 STÜCK

Rotieren Sie Ihren Clusterschlüssel, um die Sicherheitsanforderungen zu erfüllen und potenzielle Sicherheitslücken zu vermeiden. Für diesen Vorgang muss Ihr Cluster in den Wartungsmodus versetzt werden.

> **Note**
>
> Standardmäßig verschlüsselt AWS PCS das Cluster-Geheimnis mit einem Von AWS verwalteter Schlüssel. Wenn Sie es stattdessen mit einem vom Kunden verwalteten Schlüssel verschlüsseln, stellen Sie sicher, dass die Schlüsselrichtlinie der AWS PCS-Rolle Zugriff auf den Schlüssel gewährt. Andernfalls kann AWS PCS das Geheimnis nicht rotieren. Weitere Informationen finden Sie unter Verwenden Sie einen vom Kunden verwalteten Schlüssel, um das Cluster-Geheimnis zu verschlüsseln.

Voraussetzungen

- IAM-Rolle mit Genehmigung `secretsmanager:RotateSecret`
- Cluster in unserem `ACTIVE` Bundesstaat `UPDATE_FAILED`

Verfahren

1. Informieren Sie Cluster-Benutzer über das bevorstehende Wartungsfenster.
2. Versetzen Sie den Cluster in den Wartungsmodus, indem Sie alle Rechenknotengruppen auf die Kapazität 0 skalieren.

    a. Verwenden Sie die UpdateComputeNodeGroup API, um InstanceCount sowohl Min als auch Max für alle Rechenknotengruppen InstanceCount auf 0 zu setzen.

b. Warten Sie, bis alle Knoten gestoppt sind.

c. (Optional) Entleeren Sie die Scheduler-Warteschlangen mit Slurm-Befehlen, bevor Sie die Kapazität für eine reibungslose Auftragsabwicklung beenden.

> **Note**
>
> Für die Rotation sind keine laufenden Instanzen erforderlich. Wenn die Instanzen noch laufen, wenn Sie die Rotation starten, schlägt die Rotation fehl und es wird der folgende Fehler angezeigt:
>
> ```
> All instances must be terminated before you rotate a secret. Set the minimum
>  instance count to 0 to terminate active instances.
> ```

> **Note**
>
> Eine Compute-Knotengruppe kann einen Bericht erstellen`ACTIVE`, während die Instances beim Herunterskalieren noch beendet werden. Beim Hochskalieren kann sie auch Berichte erstellen, `ACTIVE` bevor eine Instanz existiert. Die Skalierung kann mehrere Minuten dauern, bis zu etwa 30 Minuten. Bei einer Compute-Knotengruppe werden die Skalierungswerte vor der Aktualisierung `get-compute-node-group` zurückgegeben`UPDATING`, und AWS PCS lehnt weitere Aktualisierungsanfragen ab, bis das Update abgeschlossen ist. Behandeln Sie das nicht `ACTIVE` als Bestätigung, dass Instanzen existieren.

3. Initiieren Sie die Rotation über Secrets Manager.
   - Konsolen-Methode:
     - Navigieren Sie zum Secrets Manager, wählen Sie Ihr Cluster-Geheimnis aus und wählen Sie Rotate Secret aus.
   - API-Methode:
     - Verwenden Sie die Secrets `rotate-secret` Manager-API.
4. Bestätigen Sie, dass die Rotation erfolgreich war.

> **Note**
>
> Der `rotate-secret` Aufruf gibt HTTP 200 zurück, auch wenn die Rotation später fehlschlägt. Das Ergebnis kommt später als separates `RotationFailed` `RotationSucceeded` OR-Ereignis. Die 200-Antwort bestätigt nur, dass AWS PCS die Anfrage akzeptiert hat, nicht, dass die Rotation erfolgreich war.

a. Führen Sie den Befehl aus `describe-secret` und bestätigen Sie, dass das `AWSCURRENT` Staging-Label auf die neue Version verschoben und diese `lastRotatedDate` aktualisiert wurde.

b. Warten Sie alternativ auf ein `RotationSucceeded` Ereignis in AWS CloudTrail.

> **Note**
>
> Verwenden Sie Versionen`list-secret-version-ids`, `describe-secret` anstatt sie zu überprüfen. `list-secret-version-ids`kann die `AWSPENDING` Version weglassen, auch wenn Sie sie angeben`--include-deprecated`.

5. Stellen Sie nach erfolgreicher Rotation die Clusterkapazität wieder her.

   a. Verwenden Sie die UpdateComputeNodeGroup API, um Knotengruppen auf die gewünschte min/max Kapazität zurückzusetzen.

   b. Für AWS PCS-managed Anmeldeknoten: Keine zusätzliche Aktion erforderlich.

   c. Für BYO-Anmeldeknoten:

      i. Stellen Sie eine Verbindung zu den Anmeldeknoten her.

      ii. Aktualisieren Sie `/etc/slurm/slurm.key` mit dem neuen Secret von Secrets Manager.

      iii. Starten Sie den Slurm Auth and Cred Kiosk Daemon (sackd) neu.

Stellen Sie eine ausstehende, gestrandete Version wieder her

Eine fehlgeschlagene Rotation kann dazu führen, dass eine Version am `AWSPENDING` Staging-Label landet und spätere Rotationen blockiert werden. Der folgende Fehler tritt auf:

```
A previous rotation isn't complete. That rotation will be reattempted.
```

`cancel-rotate-secret`Mit dem Befehl allein wird das `AWSPENDING` Staging-Label nicht entfernt, sondern der Wert wird auf gesetzt. `RotationEnabled` `false` Um die ungenutzte Version zu löschen, entfernen Sie das `AWSPENDING` Staging-Label von der ausstehenden Version:

```
aws secretsmanager update-secret-version-stage \
    --secret-id secret-arn \
    --version-stage AWSPENDING \
    --remove-from-version-id pending-version-id
```

Überprüfen Sie die Rotation

Um zu bestätigen, dass der Slurm-Controller erreichbar ist und die Knoten fehlerfrei sind, verwenden Sie `scontrol ping` und`scontrol show nodes`. Verwenden Sie es nicht`sinfo`, um einen Cluster zu überprüfen, der keine Warteschlangen hat, da in diesem Fall nichts `sinfo` angezeigt wird und keine gültige Prüfung ist.

Um zu bestätigen, dass Knoten den aktuellen Schlüssel verwenden, vergleichen Sie den SHA-256 Hash des Base64-dekodierten `AWSCURRENT` Geheimnisses mit dem Hash von `/etc/slurm/slurm.key` auf den Knoten:

```
sha256sum /etc/slurm/slurm.key
```

Vergleichen Sie auch den Node-Hash mit der `AWSPREVIOUS` Version, um zu beweisen, dass die Knoten keinen veralteten Schlüssel verwenden. Wir empfehlen, Hashes zu vergleichen, anstatt das Schlüsselmaterial auszudrucken.

Häufig gestellte Fragen zur Cluster-Secret-Rotation in AWS STCK

Hier finden Sie Antworten auf häufig gestellte Fragen zur Cluster-Rotation in AWS PCS.

Was ist ein geheimes Clustergeheimnis?

> Ein Cluster-Secret ist ein sicherer Berechtigungsnachweis, der eine sichere Kommunikation zwischen dem Slurm-Controller und den AWS PCS-Rechenknoten ermöglicht. Es dient auch als JSON Web Token (JWT) -Schlüssel für die Slurm-REST-API-Authentifizierung.

Was ist der Unterschied zwischen Cluster-Secret und JWT-Schlüssel?

In AWS PCS sind das Clustergeheimnis und der JWT-Schlüssel dieselbe Ressource, die unterschiedlichen Zwecken dient. Das Cluster-Geheimnis authentifiziert die interne Kommunikation von Slurm, während der JWT-Schlüssel Token für die REST-API-Authentifizierung signiert. Bei einer Rotation sind beide Aspekte gleichzeitig betroffen.

Wie lange dauert eine Rotation?

Der Rotationsvorgang dauert in der Regel einige Minuten. Ihr Cluster bleibt im Status AKTIV und die Abrechnung erfolgt während der Rotation normal.

Kann ich automatische Rotationen planen?

Sie können die geplante Rotation in Secrets Manager aktivieren. Die erste Version erfordert jedoch vor jeder Rotation eine manuelle Vorbereitung (Skalierung der Knotengruppen auf 0).

Funktionieren meine vorhandenen JWT-Token auch nach der Rotation noch?

Nein, bestehende JWT-Token werden nach der Rotation ungültig. Stellen Sie neue Token für REST-API-Clients aus.

Wo finde ich mein Cluster-Geheimnis?

Sie finden Ihr Cluster-Geheimnis in der Secrets Manager-Konsole oder über die AWS PCS-Konsole. Eine ausführliche Anleitung finden Sie unter Verwenden Sie AWS Secrets Manager um das Cluster-Geheimnis zu finden undVerwenden Sie AWS PCS, um das Cluster-Geheimnis zu finden.

Warum erfordert die Rotation die Skalierung von Knotengruppen auf 0?

Für die Rotation sind keine laufenden Instanzen erforderlich, um die Cluster-Stabilität während des geheimen Aktualisierungsprozesses sicherzustellen. Dadurch werden Authentifizierungskonflikte zwischen alten und neuen Geheimnissen verhindert.

Welche Compliance-Anforderungen unterstützt diese Funktion?

Diese Funktion ermöglicht es AWS PCS, branchenübliche Compliance-Standards wie HIPAA und FedRAMP zu erfüllen, die im Rahmen ihrer Sicherheitskontrollen eine regelmäßige Rotation der Anmeldeinformationen vorschreiben.

Kann ich meinen eigenen, vom Kunden verwalteten Schlüssel für das PCS Managed Secret verwenden? AWS

Ja. Sie können das Cluster-Geheimnis mit einem symmetrischen, vom `ENCRYPT_DECRYPT` Kunden verwalteten Schlüssel im selben AWS-Konto Land verschlüsseln. AWS-Region Sie

fügen den Schlüssel während des Vorgangs hinzu oder ändern ihn. AWS Secrets Manager `UpdateSecret` Es gibt keinen AWS PCS-API-Parameter für den Verschlüsselungsschlüssel des Geheimnisses.

Die Schlüsselrichtlinie muss dem AWS PCS-Dienst die Rolle (`AWSServiceRoleForPCS`) `kms:Decrypt`kms:GenerateDataKey, und `kms:DescribeKey` gewähren. Weitere Informationen finden Sie unter Verwenden Sie einen vom Kunden verwalteten Schlüssel, um das Cluster-Geheimnis zu verschlüsseln.

Kann ich den Cluster-Geheimwert selbst ändern?

Nein. AWS PCS verwaltet den geheimen Wert. Wenn Sie versuchen, das `SecretString` oder `SecretBinary` durch zu ändern AWS Secrets Manager, schlägt die Anfrage fehl. Sie können den Verschlüsselungsschlüssel (den AWS KMS Schlüssel) und die Beschreibung ändern.

Was passiert, wenn die serviceverknüpfte Rolle den Zugriff auf den Schlüssel verliert?

Rotations- und Knotenoperationen, die das Geheimnis lesen, schlagen fehl. Die zugrunde liegende Ursache ist ein Fehler mit AWS KMS Zugriffsverweigerung, der in AWS CloudTrail sichtbar ist. Die Rotation wird als fehlgeschlagene Rotation angezeigt. Der Cluster selbst bleibt bestehen. `ACTIVE`

Fehlerbehebung bei der geheimen Cluster-Rotation in AWS PCS

Die Rotation des geheimen Clusters schlägt fehl, wenn die Umgebung nicht ordnungsgemäß vorbereitet ist. Die häufigste Ursache sind aktive Instanzen in Ihrem Cluster. Um Ausfälle zu verhindern:

1. Stellen Sie für alle Knotengruppen die Kapazität 0 ein.
2. Warten Sie, bis die Knoten gestoppt sind.
3. Stellen Sie sicher, dass sich Ihr Cluster nicht in den folgenden Zuständen befindet: `CREATE_FAILED` `DELETE_FAILEDRESUMING`,`SUSPENDING`,, oder`SUSPENDED`.

Wenn die Rotation fehlschlägt:

- Es erscheint ein RotationFailed CloudTrail Ereignis
- Das Clustergeheimnis bleibt unverändert
- Einzelheiten finden Sie in CloudTrail der RotationFailed Veranstaltung
- Schließe alle Vorbereitungsschritte für eine erfolgreiche Rotation ab

AWS PCS-Rechenknotengruppen

Eine AWS PCS-Rechenknotengruppe ist eine logische Sammlung von Knoten (Amazon EC2-Instances). Diese Knoten können zur Ausführung von Rechenaufträgen sowie zum Bereitstellen eines interaktiven, Shell-basierten Zugriffs auf ein HPC-System verwendet werden. Eine Rechenknotengruppe besteht aus Regeln für die Erstellung von Knoten, einschließlich der zu verwendenden Amazon EC2-Instance-Typen, der Anzahl der auszuführenden Instances, ob Spot-Instances oder Instances verwendet werden sollen, welche Subnetze und Sicherheitsgruppen verwendet werden sollen und wie die einzelnen On-demand Instances beim Start konfiguriert werden. Wenn diese Regeln aktualisiert werden, aktualisiert AWS PCS die mit der Compute-Knotengruppe verknüpften Ressourcen entsprechend.

⚠ SMT ist auf Rechenknoten deaktiviert

AWS PCS deaktiviert Simultaneous Multithreading (SMT), auch bekannt als bei Intel-Prozessoren, Hyper-Threading auf allen Rechenknoteninstanzen beim Bootstrap. Dies ist nicht konfigurierbar. Bei SMT-capable Instanztypen ist jede vCPU einem dedizierten physischen Kern und nicht einem Hardware-Thread zugeordnet. Das bedeutet, dass die Gesamtzahl der vCPUs die Hälfte des Standardwerts für den Instanztyp beträgt, aber jede vCPU exklusiven Zugriff auf den gesamten Kern hat. Ein Instanztyp, der 96 vCPUs ankündigt, hat beispielsweise 48 nutzbare Kerne auf PCS-Rechenknoten. AWS Instance-Typen, die SMT nicht unterstützen, wie Graviton (Arm), sind nicht betroffen.
Bei den meisten rechenintensiven HPC-Workloads ist die Leistung bei deaktiviertem SMT gleichwertig oder besser. Durch die Deaktivierung von Hyperthreading werden Ressourcenkonflikte zwischen benachbarten Threads vermieden und jeder physische Kern erhält exklusiven Zugriff auf seinen Cache und seine Ausführungseinheiten. Dies ist in HPC-Umgebungen üblich.

Themen

- Erstellen einer Compute-Knotengruppe in AWS STCK
- Aktualisierung einer AWS PCS-Rechenknotengruppe
- Löschen einer Compute-Knotengruppe in AWS PCS
- Details zur Compute-Knotengruppe in AWS PCS abrufen
- Suchen nach Rechenknotengruppeninstanzen in AWS PCS

- Führen Sie benutzerdefinierte Skripts mit Node-Lifecycle-Aktionen in aus AWS STCK

Erstellen einer Compute-Knotengruppe in AWS STCK

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was zu beachten ist, wenn Sie eine Rechenknotengruppe in AWS Parallel Computing Service (AWS PCS) erstellen. Wenn Sie zum ersten Mal eine Rechenknotengruppe in AWS PCS erstellen, empfehlen wir Ihnen, dem Tutorial unter zu folgenFangen Sie an mit AWS Dienst für parallele Datenverarbeitung. Das Tutorial kann Ihnen helfen, ein funktionierendes HPC-System zu erstellen, ohne auf alle verfügbaren Optionen und Systemarchitekturen einzugehen, die möglich sind.

> **Note**
>
> Sie können benutzerdefinierte Slurm-Einstellungen für Rechenknotengruppen konfigurieren, um die Ressourcenauslastung und das Verhalten auf Knotenebene zu steuern. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK.

> **Note**
>
> Um benutzerdefinierte Skripts an bestimmten Punkten im Lebenszyklus eines Rechenknotens auszuführen — z. B. beim Mounten von Dateisystemen, Installieren von Software oder Beitreten zu einem Verzeichnisdienst —, ohne die Logik in die Benutzerdaten der Startvorlage einzubetten, verwenden Sie Node-Lifecycle-Aktionen. Weitere Informationen finden Sie unter Aktionen im Lebenszyklus eines Knotens.

> **Important**
>
> AWS PCS benötigt derzeit einen Kernel mit IPv4-Unterstützung für die Kommunikation mit lokalen Knoten, auch wenn Sie AWS PCS in einem IPv6-only Netzwerk verwenden. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Voraussetzungen

- Ausreichende Servicekontingente, um die gewünschte Anzahl von EC2-Instances in Ihrem zu starten. AWS-Region Sie können das verwenden AWS-Managementkonsole, um Ihre Servicekontingente zu überprüfen und Erhöhungen zu beantragen.
- Eine vorhandene VPC und Subnetze, die die AWS PCS-Netzwerkanforderungen erfüllen. Wir empfehlen, dass Sie sich mit diesen Anforderungen gründlich vertraut machen, bevor Sie einen Cluster für den Produktionseinsatz bereitstellen. Weitere Informationen finden Sie unter AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen. Sie können auch eine CloudFormation Vorlage verwenden, um eine VPC und Subnetze zu erstellen. AWS stellt ein HPC-Rezept für die Vorlage bereit. CloudFormation Weitere Informationen finden Sie unter https://github.com/aws-samples/aws-hpc-recipes/tree/main/recipes/net/hpc_large_scale aws-hpc-recipes auf. GitHub
- Ein IAM-Instanzprofil mit Berechtigungen zum Aufrufen der AWS `RegisterComputeNodeGroupInstance` PCS-API-Aktion und zum Zugriff auf alle anderen AWS Ressourcen, die für Ihre Knotengruppen-Instances erforderlich sind. Weitere Informationen finden Sie unter IAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung.
- Eine Startvorlage für Ihre Knotengruppen-Instances. Weitere Informationen finden Sie unter Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS.
- Um eine Compute-Knotengruppe zu erstellen, die Amazon EC2-Spot-Instances verwendet, benötigen Sie die AWSServiceRoleForEC2Spot service-verknüpfte Rolle in Ihrem. AWS-Konto Weitere Informationen finden Sie unter Amazon EC2 Spot-Rolle für AWS 2 STCK.

Erstellen Sie eine Compute-Knotengruppe in AWS STCK

Sie können eine Compute-Knotengruppe mit AWS-Managementkonsole oder erstellen AWS CLI.

AWS-Managementkonsole

So erstellen Sie Ihre Compute-Knotengruppe mithilfe der Konsole

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den Cluster aus, in dem Sie eine Compute-Knotengruppe erstellen möchten. Navigieren Sie zu Compute Node Groups und wählen Sie Create aus.
3. Geben Sie im Abschnitt zur Einrichtung der Compute-Knotengruppe einen Namen für Ihre Knotengruppe ein. Der Name darf nur alphanumerische Zeichen und Bindestriche enthalten,

bei denen zwischen Groß- und Kleinschreibung unterschieden wird. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht länger als 25 Zeichen sein. Der Name muss innerhalb des Clusters eindeutig sein.

4. Geben Sie unter Computerkonfiguration die folgenden Werte ein, oder wählen Sie sie aus:

   a. EC2-Startvorlage — Wählen Sie eine benutzerdefinierte Startvorlage aus, die für diese Knotengruppe verwendet werden soll. Startvorlagen können verwendet werden, um Netzwerkeinstellungen wie Subnetz und Sicherheitsgruppen, Überwachungskonfiguration und Speicher auf Instanzebene anzupassen. Wenn Sie noch keine Startvorlage vorbereitet haben, erfahren Sie unter, wie Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS Sie eine erstellen.

   > **Important**
   >
   > AWS PCS erstellt eine verwaltete Startvorlage für jede Rechenknotengruppe. Diese sind benannt`pcs-`*`identifier`*`-do-not-delete`. Wählen Sie diese nicht aus, wenn Sie eine Compute-Knotengruppe erstellen oder aktualisieren, da die Knotengruppe sonst nicht richtig funktioniert.

   b. Version der EC2-Startvorlage — Sie müssen eine Version Ihrer benutzerdefinierten Startvorlage auswählen. Wenn Sie die Version später ändern, müssen Sie die Compute-Knotengruppe aktualisieren, um Änderungen in der Startvorlage zu erkennen. Weitere Informationen finden Sie unter Aktualisierung einer AWS PCS-Rechenknotengruppe.

   c. AMI-ID — Wenn Ihre Startvorlage keine AMI-ID enthält oder wenn Sie den Wert in der Startvorlage überschreiben möchten, geben Sie hier eine AMI-ID ein. Beachten Sie, dass das für die Knotengruppe verwendete AMI mit AWS PCS kompatibel sein muss. Sie können auch ein Beispiel-AMI auswählen, das von bereitgestellt wird AWS. Weitere Informationen zu diesem Thema finden Sie unterAmazon Machine Images (AMIs) für AWS STK..

   d. IAM-Instanzprofil — Wählen Sie ein Instanzprofil für die Knotengruppe aus. Ein Instanzprofil gewährt der Instanz Berechtigungen für den sicheren Zugriff auf AWS Ressourcen und Dienste. Wenn Sie noch keines vorbereitet haben, können Sie die Option Basisprofil erstellen auswählen, damit AWS PCS eines für Sie mit der Mindestrichtlinie erstelltIAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung. Weitere Informationen finden Sie unter.

e. Subnetze — Wählen Sie ein oder mehrere Subnetze in der VPC aus, in der Ihr AWS PCS-Cluster bereitgestellt wird. Wenn Sie mehrere Subnetze auswählen, ist die EFA-Kommunikation zwischen Knoten nicht verfügbar, und die Kommunikation zwischen Knoten in verschiedenen Subnetzen kann zu einer erhöhten Latenz führen. Stellen Sie sicher, dass die Subnetze, die Sie hier angeben, mit den Subnetzen übereinstimmen, die Sie in der EC2-Startvorlage definiert haben. Bei On-Demand Knotengruppen mit mehr als einem Subnetz legt die Reihenfolge, in der Sie die Subnetze auflisten, die Startpriorität fest.

f. Instances — Wählen Sie einen oder mehrere Instance-Typen aus, um Skalierungsanforderungen in der Knotengruppe zu erfüllen. Alle Instanztypen müssen dieselbe Prozessorarchitektur (x86_64 oder arm64) und dieselbe Anzahl von vCPUs haben. Wenn die Instanzen über GPUs verfügen, müssen alle Instanztypen dieselbe Anzahl von GPUs haben.

g. Skalierungskonfiguration — Geben Sie die Mindest- und Höchstanzahl von Instanzen für die Knotengruppe an. Stellen Sie das Minimum (min) ein, das dem Maximum (max) für die statische Kapazität entspricht (z. B. 5 min, 5 max.). Setzen Sie das Minimum auf 0 für eine vollständig dynamische Skalierung (z. B. 0 min, 10 max). Bei einem Slurm Wert von 24,05 oder höher können Sie für gemischte Kapazitäten den Mindestwert größer als 0 und kleiner als den Höchstwert festlegen. Dadurch wird eine Basisanzahl von Instances beibehalten und bei Bedarf hochskaliert (z. B. 2 Minuten, maximal 10).

5. (Optional) Geben Sie unter Zusätzliche Einstellungen Folgendes an:

a. Kaufoption — Wählen Sie On-Demand Instances, Spot-Instances, eine unterbrechbare Kapazitätsreservierung oder einen vorhandenen Kapazitätsblock aus. Wählen Sie On-Demand, ob Sie eine On-Demand Kapazitätsreservierung (ODCR) verwenden möchten. Weitere Informationen finden Sie unter Verwendung ODCRs mit AWS PCS. Wählen Sie „Unterbrechbare Kapazitätsreservierung“, um ein gemeinsam genutztes unterbrechbares ODCR () zu verwenden. I-ODCR Weitere Informationen finden Sie unter Verwenden I-ODCRs mit AWS STK.. Wählen Sie Capacity Block, um einen vorhandenen Amazon EC2-Kapazitätsblock für die ML-Reservierung zu verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit AWS 2 STÜCK.

b. Zuweisungsstrategie — Wenn Sie die Spot-Kaufoption ausgewählt haben, können Sie angeben, wie Spot-Kapazitätspools ausgewählt werden, wenn Instances in der Knotengruppe gestartet werden. Weitere Informationen finden Sie unter Zuweisungsstrategien für Spot-Instances im Amazon Elastic Compute Cloud-

Benutzerhandbuch. Diese Option hat keine Auswirkung, wenn Sie die On-demand Kaufoption ausgewählt haben.

6. (Optional) Im Abschnitt zur Scheduler-Konfiguration können Sie Folgendes angeben:

   - Scale-down Leerlaufzeit — (Optional) Die Zeit in Sekunden, bevor ein inaktiver Knoten in dieser Knotengruppe herunterskaliert wird. Wenn nicht festgelegt, wird der Wert auf Clusterebene verwendet. Für diese Einstellung ist Slurm Version 25.11 oder höher erforderlich.

7. (Optional) Im Abschnitt „SlurmBenutzerdefinierte Einstellungen“ können Sie Parametername- und Wertepaare hinzufügen, um zusätzliche Slurm-Einstellungen zu konfigurieren. Eine vollständige Liste der unterstützten Parameter finden Sie unterBenutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.

8. (Optional) Im Abschnitt „Aktionen für den Lebenszyklus eines Nodes“ können Sie Skripts hinzufügen, die an bestimmten Punkten im Lebenszyklus eines Rechenknotens ausgeführt werden. Wählen Sie „Skript hinzufügen“. Wählen Sie die Lebenszyklusphase aus und geben Sie den Speicherort des Skripts, einen Namen, optionale Argumente, das Verhalten bei der Fehlerbehandlung und die Ausführungsrichtlinie an. Um weitere Skripts hinzuzufügen, wiederholen Sie diese Schritte. Skripts werden innerhalb einer Phase von oben nach unten ausgeführt. Um die Ausführungsreihenfolge zu ändern, wählen Sie Aktionen für ein Script aus und wählen Sie dann Nach oben oder Nach unten. Weitere Informationen finden Sie unter Aktionen im Lebenszyklus eines Knotens.

9. (Optional) Fügen Sie unter Tags beliebige Tags zu Ihrer Compute-Knotengruppe hinzu.

10. Wählen Sie Compute-Knotengruppe erstellen aus. Das Feld Status wird angezeigt, `Creating` während AWS PCS die Knotengruppe bereitstellt. Dies kann mehrere Minuten dauern.

Empfohlener nächster Schritt

- Fügen Sie Ihre Knotengruppe zu einer Warteschlange in AWS PCS hinzu, damit sie Jobs verarbeiten kann.

AWS CLI

Um Ihre Compute-Knotengruppe zu erstellen, verwenden Sie AWS CLI

Erstellen Sie Ihre Warteschlange mit dem folgenden Befehl. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:

1. *region*Ersetzen Sie durch die ID des AWS-Region , in dem Sie Ihren Cluster erstellen möchten, z. `us-east-1` B.
2. *my-cluster*Ersetzen Sie durch den Namen oder den Namen `clusterId` Ihres Clusters.
3. *my-node-group*Ersetzen Sie durch den Namen Ihrer Compute-Knotengruppe. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht länger als 25 Zeichen sein. Der Name muss innerhalb des Clusters eindeutig sein.
4. *subnet-ExampleID1*Ersetzen Sie ihn durch eine oder mehrere Subnetz-IDs aus Ihrer Cluster-VPC.
5. *lt-ExampleID1*Ersetzen Sie es durch die ID für Ihre benutzerdefinierte Startvorlage. Wenn Sie noch keine vorbereitet haben, erfahren Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS Sie unter, wie Sie eine erstellen.

   > **Important**
   >
   > AWS PCS erstellt eine verwaltete Startvorlage für jede Rechenknotengruppe. Diese sind benanntpcs-*identifier*-do-not-delete. Wählen Sie diese nicht aus, wenn Sie eine Compute-Knotengruppe erstellen oder aktualisieren, da die Knotengruppe sonst nicht richtig funktioniert.

6. *launch-template-version*Ersetzen Sie sie durch eine bestimmte Version der Startvorlage. AWS PCS ordnet Ihre Knotengruppe dieser bestimmten Version der Startvorlage zu.
7. *arn:InstanceProfile*Ersetzen Sie es durch den ARN Ihres IAM-Instanzprofils. Wenn Sie noch keines vorbereitet haben, finden Sie Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS eine Anleitung unter.
8. Ersetzen Sie *min-instances* und *max-instances* durch Ganzzahlwerte. Stellen Sie das Minimum (min) gleich dem Maximum (max) für die statische Kapazität ein (z. B. 5 min, 5 max.). Setzen Sie das Minimum auf 0 für eine vollständig dynamische Skalierung (z. B. 0 min, 10 max). Bei einem Slurm Wert von 24,05 oder höher können Sie für gemischte Kapazitäten

den Mindestwert größer als 0 und kleiner als den Höchstwert festlegen. Dadurch wird eine Basisanzahl von Instances beibehalten und bei Bedarf hochskaliert (z. B. 2 Minuten, maximal 10).

9. Durch einen *t3.large* anderen Instanztyp ersetzen. Sie können weitere Instanztypen hinzufügen, indem Sie eine Liste mit `instanceType` Einstellungen angeben. Beispiel, *--instance-configs instanceType=c6i.16xlarge instanceType=c6a.16xlarge*. Alle Instanztypen müssen dieselbe Prozessorarchitektur (x86_64 oder arm64) und dieselbe Anzahl von vCPUs haben. Wenn die Instanzen über GPUs verfügen, müssen alle Instanztypen dieselbe Anzahl von GPUs haben.

```
aws pcs create-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-name my-node-group \
    --subnet-ids subnet-ExampleID1 \
    --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
    --iam-instance-profile-arn=arn:InstanceProfile \
    --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
    --instance-configs instanceType=t3.large
```

Example— Erstellen einer Compute-Knotengruppe mit benutzerdefinierten Slurm-Einstellungen

```
aws pcs create-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-name my-node-group \
    --subnet-ids subnet-ExampleID1 \
    --custom-launch-template id=lt-ExampleID1,version='launch-template-version' \
    --iam-instance-profile-arn=arn:InstanceProfile \
    --scaling-config minInstanceCount=min-instances,maxInstanceCount=max-instance \
    --instance-configs instanceType=t3.large \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.

Example— Erstellen einer Rechenknotengruppe mit einer herunterskalierten Leerlaufzeit

```
aws pcs create-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-name my-gpu-nodes \
```

```
    --subnet-ids subnet-ExampleID1 \
    --custom-launch-template id=lt-ExampleID1,version='1' \
    --iam-instance-profile-arn=arn:InstanceProfile \
    --scaling-config minInstanceCount=0,maxInstanceCount=10 \
    --instance-configs instanceType=p4d.24xlarge \
    --slurm-configuration scaleDownIdleTimeInSeconds=300
```

Die `scaleDownIdleTimeInSeconds` Einstellung auf Gruppenebene für Rechenknoten überschreibt den Wert auf Clusterebene für Knoten in dieser Gruppe. Für diese Einstellung ist Slurm Version 25.11 oder höher erforderlich.

Es gibt mehrere optionale Konfigurationseinstellungen, die Sie dem `create-compute-node-group` Befehl hinzufügen können.

- Sie können angeben, `--amiId` ob Ihre benutzerdefinierte Startvorlage keinen Verweis auf ein AMI enthält oder ob Sie diesen Wert überschreiben möchten. Beachten Sie, dass das für die Knotengruppe verwendete AMI mit AWS PCS kompatibel sein muss. Sie können auch ein Beispiel-AMI auswählen, das von bereitgestellt wird AWS. Weitere Informationen zu diesem Thema finden Sie unterAmazon Machine Images (AMIs) für AWS STK..
- Wählen `--purchase-option` Sie hier aus, wie AWS PCS EC2-Instances für Ihre Compute-Knotengruppe kauft. On-Demand ist die Standardeinstellung.
  - `ONDEMAND`— On-Demand Instanzen verwenden. Wählen Sie diese Option auch, wenn Sie eine On-Demand Kapazitätsreservierung (ODCR) verwenden möchten. Weitere Informationen finden Sie unter Verwendung ODCRs mit AWS PCS.
  - `SPOT`— Verwenden Sie Spot-Instances. Wenn Sie Spot-Instances wählen, können Sie `--allocation-strategy` damit auch definieren, wie AWS PCS Spot-Kapazitätspools auswählt, wenn Instances in der Knotengruppe gestartet werden. Weitere Informationen finden Sie unter Zuweisungsstrategien für Spot-Instances im Amazon Elastic Compute Cloud-Benutzerhandbuch.
  - `CAPACITY_BLOCK`— Verwenden Sie einen vorhandenen Amazon EC2-Kapazitätsblock für die ML-Reservierung. Weitere Informationen finden Sie unter Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit AWS 2 STÜCK.
  - `INTERRUPTIBLE_CAPACITY_RESERVATION`— Verwenden Sie einen gemeinsam genutzten unterbrechbaren ODCR (). I-ODCR Weitere Informationen finden Sie unter Verwenden I-ODCRs mit AWS STK..
- Es ist möglich, Slurm Konfigurationsoptionen für die Knoten in der Knotengruppe mit bereitzustellen. `--slurm-configuration` Sie können die Gewichtung (Planungspriorität)

und den tatsächlichen Speicher festlegen. Knoten mit niedrigeren Gewichten haben eine höhere Priorität, und die Einheiten sind beliebig. Weitere Informationen finden Sie in der Slurm Dokumentation unter Gewicht. Realer Speicher legt die Größe (in GB) des realen Speichers auf Knoten in der Knotengruppe fest. Er soll in Verbindung mit der `CR_CPU_Memory` Option für den Cluster in AWS PCS in Ihrer Slurm Konfiguration verwendet werden. Weitere Informationen finden Sie unter RealMemory in der Slurm-Dokumentation.

- Wird verwendet`--node-lifecycle-actions`, um benutzerdefinierte Skripts an bestimmten Punkten im Lebenszyklus eines Rechenknotens auszuführen, z. B. beim Mounten von Dateisystemen, beim Installieren von Software oder beim Beitritt zu einem Verzeichnisdienst. Weitere Informationen finden Sie unter Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK. Beispiele für Befehle finden Sie unterBeispiele für Aktionen im Knotenlebenszyklus für AWS STCK.

> **Important**
>
> Das Erstellen der Compute-Knotengruppe kann mehrere Minuten dauern.

Sie können den Status Ihrer Knotengruppe mit dem folgenden Befehl abfragen. Sie können die Knotengruppe erst einer Warteschlange zuordnen, wenn ihr Status erreicht ist`ACTIVE`.

```
aws pcs get-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group
```

Aktualisierung einer AWS PCS-Rechenknotengruppe

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was bei der Aktualisierung einer AWS PCS-Rechenknotengruppe zu beachten ist. Informationen zu den benutzerdefinierten Einstellungen von Slurm finden Sie unterBenutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.

Optionen für die Aktualisierung einer AWS PCS-Rechenknotengruppe

Durch das Aktualisieren einer AWS-PCS-Rechenknotengruppe können Sie die Eigenschaften der von AWS PCS gestarteten Instances sowie die Regeln für den Start dieser Instances ändern.

Beispielsweise können Sie das AMI für Knotengruppen-Instances durch ein anderes ersetzen, auf dem eine andere Software installiert ist. Oder Sie können Sicherheitsgruppen aktualisieren, um die eingehende oder ausgehende Netzwerkkonnektivität zu ändern. Sie können auch die Skalierungskonfiguration und die bevorzugte Kaufoption ändern.

Die folgenden Einstellungen für Knotengruppen können nach der Erstellung nicht geändert werden:

- Name
- Instances

Überlegungen beim Aktualisieren einer AWS PCS-Rechenknotengruppe

Compute-Knotengruppen definieren EC2-Instanzen, die zur Verarbeitung von Aufträgen, zur Bereitstellung interaktiver Shell-Zugriffe und für andere Aufgaben verwendet werden. Sie sind oft mit einer oder mehreren AWS PCS-Warteschlangen verknüpft. Beachten Sie Folgendes, wenn Sie Ihre Compute-Knotengruppe aktualisieren, um ihr Verhalten (oder das ihrer Knoten) zu ändern:

- Änderungen an den Eigenschaften der Compute-Knotengruppe werden wirksam, wenn sich der Status der Compute-Knotengruppe von Aktualisierung zu Aktiv ändert. Neue Instanzen werden mit den aktualisierten Eigenschaften gestartet.
- Aktualisierungen, die sich nicht auf die Konfiguration bestimmter Knoten auswirken, wirken sich nicht auf laufende Knoten aus. Zum Beispiel das Hinzufügen eines Subnetzes und das Ändern der Zuweisungsstrategie.
- Bei On-Demand Knotengruppen mit mehr als einem Subnetz bestimmt die Reihenfolge, in der Sie die Subnetze auflisten, die Startpriorität. AWS PCS versucht, Instances im ersten Subnetz zu starten, und versucht, das nächste Subnetz zu starten, falls das erste Subnetz die angeforderte Kapazität nicht bereitstellen kann. Um die Priorität zu ändern, aktualisieren Sie die Knotengruppe mit den Subnetzen, die in Ihrer bevorzugten Reihenfolge aufgeführt sind. Die Änderung wirkt sich nur auf zukünftige Starts aus AWS . PCS verschiebt oder ersetzt keine laufenden Instances.
- Wenn Sie die Startvorlage für eine Compute-Knotengruppe aktualisieren, müssen Sie die Compute-Knotengruppe aktualisieren, um die neue Version zu verwenden.
- Um eine Sicherheitsgruppe zu Knoten in einer Compute-Knotengruppe hinzuzufügen oder zu entfernen, bearbeiten Sie die entsprechende Startvorlage und aktualisieren Sie die Compute-Knotengruppe. Neue Instances werden mit dem aktualisierten Satz von Sicherheitsgruppen gestartet.

- Wenn Sie eine Sicherheitsgruppe, die von einer Compute-Knotengruppe verwendet wird, direkt bearbeiten, wirkt sich dies sofort auf laufende und zukünftige Instances aus.
- Wenn Sie dem von einer Compute-Knotengruppe verwendeten IAM-Instanzprofil Berechtigungen hinzufügen oder daraus entfernen, wirkt sich dies sofort auf laufende und zukünftige Instances aus.
- Um das AMI zu ändern, das von den Instances einer Rechenknotengruppe verwendet wird, aktualisieren Sie die Rechenknotengruppe (oder ihre Startvorlage), sodass sie das neue AMI verwendet, und warten Sie, bis AWS PCS die Instances ersetzt.
- AWS PCS ersetzt die vorhandenen Instances in der Knotengruppe nach einer Aktualisierung der Knotengruppe. Wenn auf einem Knoten Jobs ausgeführt werden, dürfen diese Jobs abgeschlossen werden, bevor AWS PCS den Knoten ersetzt. Interaktive Benutzerprozesse (z. B. Instanzen auf Anmeldeknoten) werden beendet. Der Knotengruppenstatus kehrt zurück, `Active` wenn AWS PCS die Instances für den Austausch markiert. Der eigentliche Austausch erfolgt jedoch, wenn die Instances inaktiv sind.
- Wenn Sie die maximal zulässige Anzahl von Instanzen in einer Compute-Knotengruppe verringern, entfernt AWS PCS Knoten aus Slurm, um das neue Maximum zu erreichen. AWS PCS beendet laufende Instanzen, die den entfernten Slurm-Knoten zugeordnet sind. Die laufenden Jobs auf den entfernten Knoten schlagen fehl und kehren zu ihren Warteschlangen zurück.
- AWS PCS erstellt eine verwaltete Startvorlage für jede Rechenknotengruppe. Sie sind benannt`pcs-`*`identifier`*`-do-not-delete`. Wählen Sie sie nicht aus, wenn Sie eine Compute-Knotengruppe erstellen oder aktualisieren, da die Knotengruppe sonst nicht richtig funktioniert.
- Wenn Sie eine Compute-Node-Gruppe aktualisieren, um Spot für ihre Kaufoption zu verwenden, müssen Sie in Ihrem Konto über die AWSServiceRoleForEC2Spot dienstgebundene Rolle verfügen. Weitere Informationen finden Sie unter Amazon EC2 Spot-Rolle für AWS 2 STCK.

Um eine AWS PCS-Rechenknotengruppe zu aktualisieren

Sie können eine Knotengruppe mithilfe der AWS-Managementkonsole oder der AWS-CLI aktualisieren.

AWS-Managementkonsole

Um eine Rechenknotengruppe zu aktualisieren

1. Öffnen Sie die AWS PCS-Konsole unter `https://console.aws.amazon.com/pcs/home#/clusters`
2. Wählen Sie den Cluster aus, in dem Sie eine Rechenknotengruppe aktualisieren möchten.

3. Navigieren Sie zu Compute-Knotengruppen, gehen Sie zu der Knotengruppe, die Sie aktualisieren möchten, und wählen Sie dann Bearbeiten aus.
4. Aktualisieren Sie in den Abschnitten Computerkonfiguration , Zusätzliche Einstellungen und Slurm Anpassungseinstellungen alle Werte mit folgenden Ausnahmen:

   • Instanzen — Sie können die Instanzen in einer Compute-Knotengruppe nicht ändern.

   Weitere Informationen zu den benutzerdefinierten Einstellungen von Slurm finden Sie unterBenutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.
5. Im Abschnitt zur Scheduler-Konfiguration können Sie die Scale-down Leerlaufzeit aktualisieren, um einen neuen Wert (1—10000000 Sekunden) festzulegen, der den Cluster-Standard außer Kraft setzt, oder ihn löschen, um zur Einstellung auf Cluster-Ebene zurückzukehren. Für diese Einstellung ist Version 25.11 oder höher erforderlich. Slurm
6. Wählen Sie Aktualisieren aus. Während die Änderungen übernommen werden, wird im Feld Status die Meldung Aktualisierung angezeigt.

> **Important**
>
> Aktualisierungen der Compute-Knotengruppe können mehrere Minuten dauern.

AWS CLI

Um eine Compute-Knotengruppe zu aktualisieren

1. Aktualisieren Sie Ihre Compute-Knotengruppe mit dem folgenden Befehl. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:

   a. *region-code*Ersetzen Sie es durch die AWS-Region, in der Sie Ihren Cluster erstellen möchten.

   b. *my-node-group*Ersetzen Sie es durch den Namen oder `computeNodeGroupId` für Ihre Rechenknotengruppe.

   c. *my-cluster*Ersetzen Sie durch den Namen oder den `clusterId` Ihres Clusters.

```
aws pcs update-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
```

```
    --compute-node-group-identifier my-node-group
```

Example— Aktualisierung einer Compute-Knotengruppe mit benutzerdefinierten Slurm-Einstellungen

```
aws pcs update-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.

Example— Aktualisierung der Scale-Down-Leerlaufzeit für eine Rechenknotengruppe

```
aws pcs update-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-gpu-nodes \
    --slurm-configuration scaleDownIdleTimeInSeconds=600
```

Sie können eine bestehende Rechenknotengruppe aktualisieren, um die Leerlaufzeit beim Herunterskalieren `scaleDownIdleTimeInSeconds` auf Clusterebene zu überschreiben. Gültige Werte sind 1—10000000. Für diese Einstellung ist Slurm Version 25.11 oder höher erforderlich.

Example— Das Herunterskalieren der Leerlaufzeitüberschreibung aus einer Compute-Knotengruppe wird entfernt

```
aws pcs update-compute-node-group --region region \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-gpu-nodes \
    --slurm-configuration scaleDownIdleTimeInSeconds=-1
```

Stellen Sie `scaleDownIdleTimeInSeconds` diese Option auf `-1` ein, um die Knotengruppen-Override zu entfernen und zur Einstellung auf Cluster-Ebene zurückzukehren.

2. Aktualisieren Sie alle Knotengruppenparameter mit Ausnahme von. `--instance-configs` Um beispielsweise eine neue AMI-ID festzulegen, *`my-custom-ami-id`* wird pass `--amiId` `my-custom-ami-id` where durch das AMI Ihrer Wahl ersetzt.

> **Important**
>
> Die Aktualisierung der Compute-Knotengruppe kann mehrere Minuten dauern.

Sie können den Status Ihrer Knotengruppe mit dem folgenden Befehl abfragen.

```
aws pcs get-compute-node-group --region region-code \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-node-group
```

Löschen einer Compute-Knotengruppe in AWS PCS

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was zu beachten ist, wenn Sie eine Rechenknotengruppe in AWS PCS löschen.

Überlegungen beim Löschen einer Compute-Knotengruppe

Compute-Knotengruppen definieren EC2-Instances, die für die Verarbeitung von Jobs, für den interaktiven Shell-Zugriff und für andere Aufgaben verwendet werden. Sie sind häufig mit einer oder mehreren AWS PCS-Warteschlangen verknüpft. Bevor Sie eine Compute-Knotengruppe löschen, sollten Sie Folgendes beachten:

- Alle von der Compute-Knotengruppe gestarteten EC2-Instances werden beendet. Dadurch werden Jobs storniert, die auf diesen Instances ausgeführt werden, und laufende interaktive Prozesse werden beendet.
- Sie müssen die Zuordnung der Compute-Knotengruppe zu allen Warteschlangen aufheben, bevor Sie sie löschen können. Weitere Informationen finden Sie unter Aktualisierung einer AWS PCS-Warteschlange.

Löschen Sie die Compute-Knotengruppe

Sie können das AWS-Managementkonsole oder verwenden AWS CLI , um eine Compute-Knotengruppe zu löschen.

AWS-Managementkonsole

Um eine Compute-Knotengruppe zu löschen

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den Cluster der Compute-Knotengruppe aus.
3. Navigieren Sie zu Compute-Knotengruppen und wählen Sie die zu löschende Compute-Knotengruppe aus.
4. Wählen Sie Löschen aus.
5. Das Feld Status wird angezeigt`Deleting`. Das kann mehrere Minuten dauern.

> **Note**
>
> Sie können Befehle verwenden, die Ihrem Scheduler eigen sind, um zu bestätigen, dass die Compute-Knotengruppe gelöscht wurde. Verwenden Sie zum Beispiel `sinfo` or `squeue` für Slurm.

AWS CLI

Um eine Compute-Knotengruppe zu löschen

- Verwenden Sie den folgenden Befehl, um eine Compute-Knotengruppe mit diesen Ersetzungen zu löschen:
  - Ersetzen Sie *`region-code`* durch den, in dem sich AWS-Region Ihr Cluster befindet.
  - *`my-node-group`*Ersetzen Sie durch den Namen oder die ID Ihrer Compute-Knotengruppe.
  - *`my-cluster`*Ersetzen Sie durch den Namen oder die ID Ihres Clusters.

```
aws pcs delete-compute-node-group --region region-code \
       --compute-node-group-identifier my-node-group \
```

```
    --cluster-identifier my-cluster
```

Das Löschen der Compute-Knotengruppe kann mehrere Minuten dauern.

> **Note**
>
> Sie können Befehle verwenden, die Ihrem Scheduler eigen sind, um zu bestätigen, dass die Compute-Knotengruppe gelöscht wurde. Verwenden Sie zum Beispiel `sinfo` or `squeue` für Slurm.

Details zur Compute-Knotengruppe in AWS PCS abrufen

Sie können das AWS-Managementkonsole oder verwenden, AWS CLI um Details zu einer Rechenknotengruppe abzurufen, z. B. ihre Compute-Knotengruppen-ID, den Amazon-Ressourcennamen (ARN) und die Amazon Machine Image (AMI) -ID. Bei diesen Details handelt es sich häufig um erforderliche Werte für AWS PCS-API-Aktionen und -Konfigurationen.

AWS-Managementkonsole

Um Details zur Compute-Knotengruppe abzurufen

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den -Cluster.
3. Wählen Sie Compute Node Groups aus.
4. Wählen Sie im Listenbereich eine Compute-Knotengruppe aus.

AWS CLI

Um Details zur Compute-Knotengruppe abzurufen

1. Verwenden Sie die ListClustersAPI-Aktion, um Ihren Clusternamen oder Ihre Cluster-ID zu ermitteln.

```
aws pcs list-clusters
```

Beispielausgabe:

```
{
    "clusters": [
        {
            "name": "get-started-cfn",
            "id": "pcs_abc1234567",
            "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567",
            "createdAt": "2025-04-01T20:11:22+00:00",
            "modifiedAt": "2025-04-01T20:11:22+00:00",
            "status": "ACTIVE"
        }
    ]
}
```

2. Verwenden Sie die ListComputeNodeGroupsAPI-Aktion, um die Compute-Knotengruppen in einem Cluster aufzulisten.

```
aws pcs list-compute-node-groups --cluster-identifier cluster-name-or-id
```

Beispiel für einen Aufruf:

```
aws pcs list-compute-node-groups --cluster-identifier get-started-cfn
```

Beispielausgabe:

```
{
    "computeNodeGroups": [
        {
            "name": "compute-1",
            "id": "pcs_abc123abc1",
            "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
computenodegroup/pcs_abc123abc1",
            "clusterId": "pcs_abc1234567",
            "createdAt": "2025-04-01T20:19:25+00:00",
            "modifiedAt": "2025-04-01T20:19:25+00:00",
            "status": "ACTIVE"
        },
        {
            "name": "login",
            "id": "pcs_abc456abc7",
            "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
computenodegroup/pcs_abc456abc7",
```

```
            "clusterId": "pcs_abc1234567",
            "createdAt": "2025-04-01T20:19:31+00:00",
            "modifiedAt": "2025-04-01T20:19:31+00:00",
            "status": "ACTIVE"
        }
    ]
}
```

3. Verwenden Sie die GetComputeNodeGroupAPI-Aktion, um zusätzliche Details für eine Compute-Knotengruppe abzurufen.

```
aws pcs get-compute-node-group --cluster-identifier cluster-name-or-id --
compute-node-group-identifier compute-node-group-name-or-id
```

Beispiel für einen Aufruf:

```
aws pcs get-compute-node-group --cluster-identifier get-started-cfn --compute-
node-group-identifier compute-1
```

Beispielausgabe:

```
{
    "computeNodeGroup": {
        "name": "compute-1",
        "id": "pcs_abc123abc1",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_abc1234567/
computenodegroup/pcs_abc123abc1",
        "clusterId": "pcs_abc1234567",
        "createdAt": "2025-04-01T20:19:25+00:00",
        "modifiedAt": "2025-04-01T20:19:25+00:00",
        "status": "ACTIVE",
        "amiId": "ami-0123456789abcdef0",
        "subnetIds": [
            "subnet-abc012345789abc12"
        ],
        "purchaseOption": "ONDEMAND",
        "customLaunchTemplate": {
            "id": "lt-012345abcdef01234",
            "version": "1"
        },
        "iamInstanceProfileArn": "arn:aws:iam::111122223333:instance-profile/
AWSPCS-get-started-cfn-us-east-1",
```

```
        "scalingConfiguration": {
            "minInstanceCount": 0,
            "maxInstanceCount": 4
        },
        "instanceConfigs": [
            {
                "instanceType": "c6i.xlarge"
            }
        ]
    }
}
```

Suchen nach Rechenknotengruppeninstanzen in AWS PCS

Jede AWS PCS-Compute-Knotengruppe kann EC2-Instances mit gemeinsam genutzten Konfigurationen starten. Sie können EC2-Tags verwenden, um Instances in einer Compute-Knotengruppe im AWS-Managementkonsole oder mit dem zu finden. AWS CLI

AWS-Managementkonsole

Um Ihre Compute-Knotengruppen-Instances zu finden

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den -Cluster.
3. Wählen Sie Compute Node Groups aus.
4. Suchen Sie die ID für die Login-Knotengruppe, die Sie erstellt haben.
5. Navigieren Sie zur EC2-Konsole und wählen Sie Instances aus.
6. Suchen Sie nach den Instances mit dem folgenden Tag. *node-group-id*Ersetzen Sie es durch die ID (nicht den Namen) Ihrer Rechenknotengruppe.

   ```
   aws:pcs:compute-node-group-id=node-group-id
   ```

7. (Optional) Sie können den Wert von Instance state im Suchfeld ändern, um nach Instances zu suchen, die gerade konfiguriert werden oder die kürzlich beendet wurden.
8. Suchen Sie die Instanz-ID und IP-Adresse für jede Instanz in der Liste der markierten Instanzen.

AWS CLI

Verwenden Sie die folgenden Befehle, um Ihre Knotengruppen-Instances zu finden. Nehmen Sie vor dem Ausführen der Befehle die folgenden Ersetzungen vor:

- *region-code*Ersetzen Sie es durch das AWS-Region Ihres Clusters. Beispiel: `us-east-1`
- *node-group-id*Ersetzen Sie durch die ID (nicht den Namen) Ihrer Rechenknotengruppe. Informationen zur ID einer Compute-Knotengruppe finden Sie unterDetails zur Compute-Knotengruppe in AWS PCS abrufen.
- `running`Ersetzen Sie diese durch andere Instanzstatus, z. B. durch `pending` oder`terminated`, um EC2-Instances in anderen Status zu finden.

```
aws ec2 describe-instances \
     --region region-code --filters \
     "Name=tag:aws:pcs:compute-node-group-id,Values=node-group-id" \
     "Name=instance-state-name,Values=running" \
     --query 'Reservations[*].Instances[*].
{InstanceID:InstanceId,State:State.Name,PublicIP:PublicIpAddress,PrivateIP:PrivateIpAddress}
```

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht. Der Wert von `PublicIP` ist, `null` wenn sich die Instance in einem privaten Subnetz befindet.

```
[
    [
        {
            "InstanceID": "i-0123456789abcdefa",
            "State": "running",
            "PublicIP": "18.189.32.188",
            "PrivateIP": "10.0.0.1"
        }
    ]
]
```

> **Note**
>
> Wenn Sie damit `describe-instances` rechnen, eine große Anzahl von Instances zurückzugeben, müssen Sie Optionen für mehrere Seiten verwenden. Weitere

Informationen finden Sie DescribeInstancesin der Amazon Elastic Compute Cloud API-Referenz.

Führen Sie benutzerdefinierte Skripts mit Node-Lifecycle-Aktionen in aus AWS STCK

Bei Aktionen im Lebenszyklus eines Knotens werden benutzerdefinierte Skripts an definierten Punkten im Lebenszyklus eines Rechenknotens ausgeführt. Verwenden Sie sie, um Dateisysteme zu mounten, Verzeichnisdiensten beizutreten, Software zu installieren oder die Überwachung einzurichten. Sie definieren Skripts in der Konfiguration der Compute-Node-Gruppe und verwenden sie in Rechenknotengruppen und Clustern wieder.

Eine Node-Lifecycle-Aktion ist ein Skript, das AWS PCS automatisch an einem bestimmten Punkt im Lebenszyklus eines Rechenknotens ausführt. Sie definieren Skripts in Ihrer Compute-Node-Gruppenkonfiguration: wo sich jedes Skript befindet (Amazon S3 oder HTTPS), welche Argumente übergeben werden müssen, in welcher Phase es ausgeführt wird, wann es beim Neustart erneut ausgeführt wird und was zu tun ist, wenn es fehlschlägt. AWS PCS ruft das Skript ab, speichert es auf der Instance, führt es als `root` aus und schreibt seine Ausgabe in eine dedizierte Protokolldatei.

Aktionen im Knotenlebenszyklus bieten:

- Wiederverwendbarkeit — Verweisen Sie aus vielen Compute-Knotengruppen mit unterschiedlichen Argumenten auf dasselbe Skript.
- Per-script Fehlerbehandlung — Beenden Sie den Knoten, stoppen Sie die Sequenz oder fahren Sie bei einem Ausfall fort.
- API-native Konfiguration — Definieren und aktualisieren Sie Lebenszyklusaktionen über`UpdateComputeNodeGroup`. Durch Änderungen an der Konfiguration werden vorhandene Knoten entleert und ersetzt. Weitere Informationen finden Sie unter Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK.

Lebenszyklusphasen

AWS PCS unterstützt derzeit zwei Stufen. In jeder Phase werden die Skripts in der Reihenfolge ausgeführt, in der Sie sie definieren. Standardmäßig wird ein Skript nur beim ersten Start ausgeführt; setzen Sie es auf`executionPolicy`, `EVERY_BOOT` um es auch beim Neustart erneut auszuführen. Weitere Informationen finden Sie unter Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK.

Stage	Wenn es läuft	Verfügbare Dienste	Typischer Gebrauch
Knoten bootstrapped () `nodeBootstrapped`	Wird ausgeführt, nachdem AWS PCS die Knoteneinrichtung abgeschlossen hat und vor dem Start `slurmd`.	Netzwerk und Amazon S3 sind aktiv und SSH, SSM, NTP (Chrony) und Cron sind für das Debuggen aktiv. AWS PCS-managed Die Einrichtung ist abgeschlossen (Slurm konfiguriert, Hyper-Threading deaktiviert), sodass Sie sich darauf verlassen können. Die Befehle des Slurm-Clients sind noch nicht verfügbar.	Aufgaben, die abgeschlossen werden müssen, bevor der Knoten Jobs akzeptiert, wie z. B. das Mounten von gemeinsam genutztem Speicher, das Konfigurieren des Netzwerks oder das Installieren von Softwarepaketen.
Knoten bereit (`nodeReady`)	Läuft nach dem `slurmd` Start und der Knoten registriert sich beim Slurm-Controller.	Alles auf der `nodeBootstrapped` Bühne, plus Slurm-Client-Befehle (`sinfo`,`scontrol`).	Aufgaben, für die Slurm ausgeführt werden muss.

⚠ **`NodeReady` und das** Job-Dispatch-Fenster

`nodeReady`Skripte werden nach dem `slurmd` Start ausgeführt. Es gibt also ein kurzes Fenster, in dem der Scheduler möglicherweise einen Job an den Knoten sendet, bevor Ihre Skripts fertig sind. `nodeReady` Wenn ein Skript abgeschlossen sein muss, bevor ein Job ausgeführt wird:

- Verschiebe es auf die `nodeBootstrapped` Bühne, wenn es keine Slurm-Befehle benötigt.
- Benutze einen Slurm-Prolog.

Wie funktioniert die Hinrichtung

Beim ersten Start:

1. Der Knoten wird gestartet und der AWS PCS-Agent ruft an`RegisterComputeNodeGroupInstance`.
2. Der Agent speichert die Lebenszykluskonfiguration im Cache und lädt die Skripts herunter.
3. Die AWS PCS-Konfigurationsphase läuft (Slurm ist konfiguriert, Hyper-Threading ist deaktiviert usw.).
4. Node-Bootstrap-Skripte werden in der Reihenfolge ausgeführt, in der Sie sie definiert haben.
5. `slurmd`startet und der Knoten registriert sich beim Slurm-Controller. Zu diesem Zeitpunkt ist der Knoten verfügbar, um Jobs anzunehmen.
6. Node Ready-Skripts werden in der Reihenfolge ausgeführt, in der Sie sie definiert haben. Diese Skripts werden ausgeführt, nachdem der Knoten für die Annahme von Aufträgen verfügbar ist, sodass der Scheduler möglicherweise einen Job versendet, bevor die Ausführung abgeschlossen ist (siehe Lebenszyklusphasen).

Beim Neustart wird dieselbe `nodeBootstrapped` → `slurmd` → `nodeReady` Sequenz ausgeführt. Nur Skripte, die als `EVERY_BOOT` erneut ausgeführt gekennzeichnet sind. Skripts, die auf `FIRST_BOOT_ONLY` (Standardeinstellung) gesetzt sind, werden übersprungen. Der `scriptCachingPolicy` steuert, ob Skripts erneut heruntergeladen werden. Weitere Informationen finden Sie unter Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK.

Beschränkungen der Anzahl von Skripten pro Stufe, der Länge von Argument- und Skriptnamen sowie der maximalen Skriptgröße finden Sie UpdateComputeNodeGroup in der AWS PCS-API-Referenz.

Themen

- Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK
- Beispiele für Aktionen im Knotenlebenszyklus für AWS STCK
- Verwenden Sie AWS-gepflegte Skripte für Node-Lifecycle-Aktionen in AWS STCK
- Bewährte Methoden für Node-Lifecycle-Aktionen in AWS STCK
- Problembehandlung bei Aktionen im Knotenlebenszyklus in AWS STCK

Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK

Sie definieren Aktionen für den Knotenlebenszyklus in Ihrer Compute-Knotengruppenkonfiguration. In diesem Thema wird beschrieben, wie Sie ein Skript definieren, steuern, ob es beim Neustart erneut ausgeführt wird, es speichern, Argumente übergeben, Fehler behandeln, es zwischenspeichern, seine Integrität überprüfen und seine Protokolle lesen.

> ⚠ Anforderung an die Agentenversion
>
> Für Aktionen im Knotenlebenszyklus ist die AWS PCS-Agent-Version 1.5.0-1 oder höher erforderlich. Wenn Ihre Rechenknoten ein benutzerdefiniertes AMI mit einer älteren Agentenversion verwenden, aktualisieren Sie den Agenten, bevor Sie Lebenszyklusaktionen konfigurieren. Weitere Informationen finden Sie unter AWS PCS-Agent-Versionen.

Wie definiert man ein Skript

Jedes Skript hat einen`name`, ein`scriptSource`, optionale`arguments`, eine `onError` Richtlinie und ein`executionPolicy`. Die Felder Standort und Integrität sind unter gruppiert`scriptSource`.

```
{
  "name": "My script",
  "scriptSource": {
    "scriptLocation": "s3://my-bucket/my-script.sh",
    "s3VersionId": "optional, S3 only",
    "checksum": "optional 64-char SHA-256 hex"
  },
  "arguments": ["arg1", "arg2"],
  "onError": "TERMINATE",
  "executionPolicy": "FIRST_BOOT_ONLY"
}
```

Fügen Sie einer Compute-Knotengruppe Lebenszyklus-Aktionen hinzu

Sie können Lebenszyklusaktionen hinzufügen, wenn Sie eine Compute-Knotengruppe erstellen oder aktualisieren. Verwenden Sie das AWS-Managementkonsole oder das AWS CLI.

AWS-Managementkonsole

Um Lebenszyklusaktionen mithilfe der Konsole hinzuzufügen

1. Öffnen Sie die AWS PCS-Konsole.
2. Öffnen Sie die Seite zum Erstellen oder Aktualisieren einer Compute-Knotengruppe.
3. Wählen Sie im Abschnitt Aktionen zum Node-Lebenszyklus die Option Skript hinzufügen aus.
4. Wählen Sie die Skriptquelle aus:
   a. Aus der AWS PCS-Skriptbibliothek hinzufügen — Verweisen Sie auf ein AWS verwaltetes Skript. Weitere Informationen finden Sie unter Verwenden Sie AWS-gepflegte Skripte für Node-Lifecycle-Aktionen in AWS STCK.
   b. Über S3 hinzufügen — Referenzieren Sie ein Skript anhand seiner Amazon S3-URI.
   c. Über HTTPS hinzufügen — Referenzieren Sie ein Skript anhand seiner HTTPS-URL.
5. Wählen Sie die Node-Stufe aus.
6. Geben Sie für ein Skript, das Sie über S3 oder HTTPS hinzufügen, die folgenden Einstellungen an:
   a. Speicherort des Skripts — Geben Sie den S3-URI oder die HTTPS-URL für das Skript ein.
   b. Name — Geben Sie einen Namen für das Skript ein.
   c. (Optional) Prüfsumme — Geben Sie eine SHA-256 Prüfsumme ein, um die Integrität des Skripts zu überprüfen. Weitere Informationen finden Sie unter Integrität des Skripts (Prüfsummen).
   d. (Optional) Geben Sie für ein Skript, das Sie über S3 hinzufügen, eine Version ein, die auf eine bestimmte Version des Objekts verweist.
7. Geben Sie für alle Skriptquellen die folgenden Einstellungen an:
   a. Verhalten bei der Fehlerbehandlung — Wählen Sie `TERMINATESTOP_SEQUENCE`, oder`CONTINUE`.
   b. Ausführungsrichtlinie — Wählen Sie `FIRST_BOOT_ONLY` oder`EVERY_BOOT`.
   c. (Optional) Argumente — Fügen Sie Argumente hinzu, die an das Skript übergeben werden sollen.
8. Um weitere Skripts hinzuzufügen, wiederholen Sie diese Schritte.

9. Skripts werden innerhalb einer Phase von oben nach unten ausgeführt. Um die Ausführungsreihenfolge zu ändern, wählen Sie Aktionen für ein Script aus und wählen Sie dann Nach oben oder Nach unten.
10. Wählen Sie für Caching-Richtlinie die Option `CACHE_ONCE` oder `REFRESH_ON_REBOOT` aus. Die Caching-Richtlinie gilt für alle Skripts. Weitere Informationen finden Sie unter Zwischenspeichern und Aktualisierungen von Skripten.

AWS CLI

Verwenden Sie den `--node-lifecycle-actions` Parameter mit dem `update-compute-node-group` Befehl `create-compute-node-group` or. Die Parameterstruktur finden Sie NodeLifecycleActionsRequest in der AWS PCS-API-Referenz. Beispiele für Befehle finden Sie unterBeispiele für Aktionen im Knotenlebenszyklus für AWS STCK.

Steuern des Neustartverhaltens

Wird `executionPolicy` pro Skript festgelegt, um zu steuern, ob es beim Neustart erneut ausgeführt wird.

Wert	Behavior	Wann sollte dies verwendet werden?
`FIRST_BOOT_ONLY` (Standard)	Führen Sie das Skript einmal aus, beim ersten Start des Knotens. Überspringen Sie es bei jedem nachfolgenden Neustart.	One-time Einrichtung, z. B. das Installieren von Paketen, das Beitreten zu einer Domäne oder das Initialisieren des Speichers . Sie müssen diese Skripts nicht idempotent machen.
`EVERY_BOOT`	Führen Sie das Skript beim ersten Start und bei jedem Neustart aus.	Konfiguration, die nach einem Neustart erneut angewendet werden muss, z. B. beim erneuten Mounten eines kurzlebigen Dateisystems oder bei der Wiederherstellung des Knotenzustands.

Skripts sind so eingestellt, dass sie einmal `FIRST_BOOT_ONLY` ausgeführt werden und beim Neustart übersprungen werden. Skripte sind so eingerichtet, dass sie bei jedem Systemstart `EVERY_BOOT` ausgeführt werden und idempotent sein sollten (es ist sicher, dass sie mehrmals ausgeführt werden und dasselbe Ergebnis haben).

Speicherorte von Skripten

Sie können Skripts in Amazon S3 speichern oder über HTTPS bereitstellen. Skripte können nicht inline über die API hochgeladen werden, und jedes Skript darf nicht größer als 2 MiB sein — der Agent lehnt jedes Skript ab, das diese Größe überschreitet.

- Amazon S3 (**`s3://`*`bucket`*`/`*`key`**) — Das Instanz-IAM-Profil muss `s3:GetObject` auf dem Objekt vorhanden sein. Mit einem S3-Gateway-VPC-Endpunkt können Knoten in privaten Subnetzen Skripts über den VPC-Endpunkt abrufen. Pingen Sie eine bestimmte Version mit `scriptSource.s3VersionId` (nur S3-Standorte) an.
- HTTPS (**`https://`*`hostname`*`/`*`path`**) — Der Host muss öffentlich lesbar sein (keine Authentifizierung) und der Knoten benötigt ausgehenden Internetzugang (Internet-Gateway, NAT-Gateway oder HTTP-Proxy). Dies ist nützlich für Skripte, die auf GitHub oder anderen öffentlichen Repositorys gehostet werden.

Externer Speicher bietet Versionskontrolle, Überprüfbarkeit und teamübergreifende Wiederverwendung. Inline- oder hochgeladene Skripts werden nicht unterstützt.

Übergabe von Argumenten an Skripts

Übergeben Sie Argumente als geordnetes Array. Sie erreichen Ihr Skript als positionale Befehlszeilenparameter.

```
arguments: ["fs-12345678", "/shared", "nfs4"]
# script.sh fs-12345678 /shared nfs4   ($1=fs-12345678, $2=/shared, $3=nfs4)
```

Der Agent exportiert Umgebungsvariablen in jedes Skript. Diese Variablen enthalten Cluster- und Knotenmetadaten.

Variable	Beschreibung
`PCS_CLUSTER_ID` / `PCS_CLUSTER_NAME`	Cluster-ID//Name

Variable	Beschreibung
`PCS_COMPUTE_NODE_GROUP_ID` / `PCS_COMPUTE_NODE_GROUP_NAME`	Gruppenkennung des Knotens berechnen//Name
`PCS_NODE_ID`	Knoten-ID
`PCS_IS_FIRST_BOOT`	1beim ersten Start des Knotens, `0` bei jedem nachfolgenden Neustart. Verwenden Sie es, um das Verhalten in `EVERY_BOOT` Skripten zu verzweigen.

```
#!/usr/bin/env bash
echo "Configuring node $PCS_NODE_ID in cluster $PCS_CLUSTER_NAME"
```

Fehlerbehandlung

Jedes Skript hat ein on`Error` Feld, das steuert, was passiert, wenn ein Skript ungleich Null beendet wird oder durch ein Signal beendet wird.

Wert	Behavior	Wann sollte dies verwendet werden?
`TERMINATE` (Standard)	Markieren Sie den Knoten als ausgefallen und beenden Sie ihn.	Kritische Konfiguration (Speichermounts, Active Directory-Join). Verhindert, dass für defekte Knoten bezahlt wird.
`STOP_SEQUENCE`	Stoppen Sie die verbleibenden Skripts in der Phase; lassen Sie den Knoten laufen.	Debugging — überprüfen Sie die Instanz nach einem Ausfall.
`CONTINUE`	Protokollieren Sie den Fehler und führen Sie das nächste Skript aus.	Optionale oder nach besten Kräften ausgeführte Aufgaben.

Ein Skript schlägt fehl, wenn es ungleich Null beendet wird oder wenn es durch ein Signal beendet wird (z. B.). `SIGSEGV` Skripts werden nicht erneut versucht. Wenn bei einem Vorgang vorübergehende Fehler auftreten können, fügen Sie dem Skript eine Wiederholungslogik hinzu. Das Abrufen (Herunterladen) des Skripts wird jedoch erneut versucht — bis zu 3 Versuche mit exponentiellem Backoff (maximal etwa 17 Sekunden) — bevor das Verhalten eintritt. `onError`

Zwischenspeichern und Aktualisierungen von Skripten

Der Agent lädt jedes Skript beim ersten Start auf die Instanz herunter und speichert es lokal. Zwei Felder steuern das Verhalten beim Neustart: `scriptCachingPolicy` steuert den erneuten Download und `executionPolicy` steuert die erneute Ausführung.

- **`CACHE_ONCE`**(Standard) — Beim ersten Start einmal herunterladen; niemals erneut abrufen. Das Verhalten ist bei allen Neustarts identisch. Für die Aktualisierung des Skriptinhalts ist ein Austausch der Instanz erforderlich.
- **`REFRESH_ON_REBOOT`**— Re-download Bei jedem Neustart wird der Cache überschrieben. Auf diese Weise können Sie Skriptkorrekturen durch einen Neustart versenden. Es erfordert Netzwerkzugriff bei jedem Start. Schlägt eine Aktualisierung fehl, wird der Download als Fehler beim Abrufen behandelt und das `onError` Verhalten des Skripts wird übernommen (es gibt kein Fallback auf die zwischengespeicherte Kopie). Ein aktualisiertes Skript wird nur dann beim Neustart erneut ausgeführt, wenn dies der Fall ist. `executionPolicy` `EVERY_BOOT`

Die Lebenszykluskonfiguration (welche Skripte ausgeführt werden, ihre Argumente, Fehlerbehandlung und Ausführungsrichtlinie) ist für jede Instanz immer unveränderlich. Eine Änderung `UpdateComputeNodeGroup` wirkt sich nur auf neue Instanzen aus und löst die `DRAIN` Strategie aus, sodass laufende Jobs beendet werden, bevor die Knoten ersetzt werden.

Integrität des Skripts (Prüfsummen)

Geben Sie optional eine SHA-256 `checksum` in einem Skript als `scriptSource` 64-stellige Hexadezimalzeichenfolge an. Der Agent überprüft es beim Herunterladen; eine Nichtübereinstimmung wird als Download-Fehler behandelt und ausgelöst. `onError` Wir empfehlen eine Prüfsumme für die Produktion, insbesondere für Skripte aus gemeinsam genutzten oder externen Quellen.

```
sha256sum mount-efs.sh   # use the 64-char hex hash as the checksum value
```

Loggen und Debuggen

Jedes Skript schreibt in seine eigene Protokolldatei, und der Agent führt sein eigenes Betriebsprotokoll.

```
# agent: download, caching, checksum, orchestration
/var/log/amazon/pcs/lifecycle/actions/executor.log
# each script's stdout/stderr
/var/log/amazon/pcs/lifecycle/actions/<stage>/<script-name>.log
```

Der Protokolldateiname verwendet den Skriptnamen, den Sie definiert haben. Leerzeichen werden beibehalten. Beispielsweise `Mount EFS home directory` schreibt ein Skript mit dem Namen Writes to`Mount EFS home directory.log`. Stellen Sie eine Verbindung mit SSH oder AWS Systems Manager Session Manager her, um sie zu lesen — beide sind `nodeBootstrapped` stufenweise verfügbar. Da Knoten, bei denen ein Fehler auftritt, ersetzt `TERMINATE` werden, leiten Sie die Protokolle nach der Beendigung zum Debuggen an andere Instanzen weiter: Fügen Sie ein Node-Bootstrap-Skript hinzu, das den CloudWatch Amazon-Agenten so konfiguriert, dass er das Lifecycle-Log-Verzeichnis an Amazon Logs sendet. CloudWatch Das AWS-maintened-Skript erledigt dies. `configure-cloudwatch-logs.sh` Weitere Informationen finden Sie unter Verwenden Sie AWS-gepflegte Skripte für Node-Lifecycle-Aktionen in AWS STCK.

Beispiele für Aktionen im Knotenlebenszyklus für AWS STCK

In den folgenden Beispielen wird der verwendet AWS CLI , um Aktionen im Knotenlebenszyklus zu konfigurieren. Sie können Lebenszyklusaktionen definieren`create-compute-node-group`, wenn Sie eine Rechenknotengruppe mit erstellen oder sie zu einer vorhandenen Rechenknotengruppe mit hinzufügen oder ändern`update-compute-node-group`. Der `--node-lifecycle-actions` Wert ist für beide Befehle derselbe.

Mehrere Skripte in verschiedenen Stufen

In diesem Beispiel werden Skripts aus beiden Lebenszyklusphasen mit unterschiedlichen Ausführungsrichtlinien und Fehlerverhalten kombiniert.

```
aws pcs create-compute-node-group --region region \
  --cluster-identifier my-cluster \
  --compute-node-group-identifier my-cng \
  --subnet-ids subnet-ExampleID1 \
  --custom-launch-template id=lt-ExampleID1,version='1' \
  --iam-instance-profile-arn=arn:InstanceProfile \
```

```
  --scaling-config minInstanceCount=0,maxInstanceCount=10 \
  --instance-configs instanceType=t3.large \
  --node-lifecycle-actions '{
    "stages": {
      "nodeBootstrapped": [
        { "name": "Mount FSx Lustre", "scriptSource": { "scriptLocation": "s3://my-
bucket/mount-fsx.sh" }, "arguments": ["fs-0abc123", "/scratch"], "executionPolicy":
 "EVERY_BOOT" },
        { "name": "Join Active Directory", "scriptSource": { "scriptLocation":
 "s3://my-bucket/configure-ad.sh" }, "arguments": ["ad.example.com", "EXAMPLE"],
 "executionPolicy": "FIRST_BOOT_ONLY" }
      ],
      "nodeReady": [
        { "name": "Configure CloudWatch logging", "scriptSource": { "scriptLocation":
 "s3://my-bucket/setup-cloudwatch.sh" }, "arguments": ["/aws/pcs/my-cluster"],
 "onError": "CONTINUE" }
      ]
    }
  }'
```

Gemischtes Fehlerverhalten

In diesem Beispiel werden kritische und optionale Skripts in derselben Phase verwendet. Der erste beendet den Knoten bei einem Ausfall; der zweite fährt fort.

```
aws pcs create-compute-node-group --region region \
  --cluster-identifier my-cluster \
  --compute-node-group-identifier my-cng \
  --subnet-ids subnet-ExampleID1 \
  --custom-launch-template id=lt-ExampleID1,version='1' \
  --iam-instance-profile-arn=arn:InstanceProfile \
  --scaling-config minInstanceCount=0,maxInstanceCount=10 \
  --instance-configs instanceType=t3.large \
  --node-lifecycle-actions '{
    "stages": {
      "nodeBootstrapped": [
        { "name": "Install required packages", "scriptSource": { "scriptLocation":
 "s3://my-bucket/install-packages.sh" }, "onError": "TERMINATE" },
        { "name": "Install optional tools", "scriptSource": { "scriptLocation":
 "s3://my-bucket/install-optional.sh" }, "onError": "CONTINUE" }
      ]
    }
  }'
```

Überprüfung der Integrität mit einer Prüfsumme

Fügt eine SHA-256 Prüfsumme hinzu, sodass der Agent ein Skript ablehnt, wenn sein Inhalt nicht übereinstimmt.

```
aws pcs update-compute-node-group \
  --cluster-identifier my-cluster \
  --compute-node-group-identifier my-cng \
  --node-lifecycle-actions '{
    "stages": {
      "nodeBootstrapped": [
        {
          "name": "Mount EFS",
          "scriptSource": {
            "scriptLocation": "s3://my-bucket/mount-efs.sh",
            "checksum":
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
          },
          "arguments": ["fs-12345678", "/shared"]
        }
      ]
    }
  }'
```

Aktivieren Sie die Skriptaktualisierung beim Neustart

Setzt `scriptCachingPolicy` auf`REFRESH_ON_REBOOT`. Bei diesem `UpdateComputeNodeGroup` Aufruf handelt es sich um eine Konfigurationsänderung. Er entleert und ersetzt bestehende Instanzen. Nach Abschluss der Ersetzung laden die Instanzen bei jedem Neustart erneut Skripts herunter. Anschließend können Sie den Skriptinhalt in Amazon S3 aktualisieren, und die Instances übernehmen die Änderung bei ihrem nächsten Neustart ohne weiteren API-Aufruf.

```
aws pcs update-compute-node-group \
  --cluster-identifier my-cluster \
  --compute-node-group-identifier my-cng \
  --node-lifecycle-actions '{
    "stages": { "nodeBootstrapped": [ { "name": "Mount EFS", "scriptSource":
 { "scriptLocation": "s3://my-bucket/mount-efs.sh" }, "arguments": ["fs-12345678", "/
shared"], "executionPolicy": "EVERY_BOOT" } ] },
    "scriptCachingPolicy": "REFRESH_ON_REBOOT"
  }'
```

Migration von ParallelCluster AWS-Bootstrap-Aktionen

Viele vorhandene AWS ParallelCluster Bootstrap-Skripte funktionieren unverändert mit AWS PCS-Lebenszyklusaktionen. AWS PCS bietet eine skriptspezifische Fehlerbehandlung (`TERMINATESTOP_SEQUENCE`, oder`CONTINUE`) und die Neustartsteuerung (`FIRST_BOOT_ONLY`oder). `EVERY_BOOT` Die AWS PCS-Standardeinstellung ist`FIRST_BOOT_ONLY`. Um dem ParallelCluster Verhalten zu entsprechen (bei jedem Start ausführen), setzen Sie `executionPolicy` den Wert auf`EVERY_BOOT`.

AWS ParallelCluster	AWS STK.
`OnNodeStart`	`nodeBootstrapped` (läuft nach der AWS PCS-Konfigurationsphase, vor`slurmd`)
`OnNodeConfigured`	`nodeReady` (läuft, nachdem sich der Knoten beim Slurm-Controller registriert hat)
`OnNodeUpdated`	Kein AWS PCS-Äquivalent

Verwenden Sie AWS-gepflegte Skripte für Node-Lifecycle-Aktionen in AWS STCK

AWS veröffentlicht gepflegte Skripte für allgemeine Lebenszyklusaktionen. Sie verweisen direkt aus Ihrer Compute-Knotengruppenkonfiguration auf ein verwaltetes Skript. Die Skripte sind idempotent und können sicher bei jedem Start ausgeführt werden.

AWS verteilt die Skripte jeweils aus einem öffentlichen, AWS verwalteten Amazon S3-Bucket. AWS-Region Sie verweisen auf ein Skript anhand seiner Amazon S3-URI. Sie verweisen auf ein verwaltetes Skript auf die gleiche Weise, wie Sie auf Ihr eigenes Skript verweisen. Sie können die Skripts direkt aus dem AWS-managed-Bucket referenzieren.

Verfügbare Skripte

Script	Beschreibung	Erforderliche Pakete
`mount-efs.sh`	Mounten Sie ein Amazon EFS-Datei system an einem bestimmten Pfad und speichern es auch nach Neustarts.	`amazon-efs-utils` , `nfs-utils`
`mount-fsx-lustre.sh`	Mounten Sie ein Amazon FSx for Lustre-Dateisystem an einem bestimmten Pfad und speichern es auch bei Neustarts.	`lustre-client`
`configure-efs-homes.sh`	Mounten Sie ein Amazon EFS-Datei system als Basisverzeichnis und konfiguri ert PAM so, dass das Home-Verz eichnis jedes Benutzers bei der ersten Anmeldung erstellt wird.	`amazon-efs-utilsnfs-utils` , und `mkhomedir` PAM-Paket e (siehe die Anforderu ngen pro Skript)
`configure-ldaps.sh`	Konfiguriert SSSD für die LDAP- oder LDAPS-Authentifizierung mit Unterstüt zung für authentifizierte Bindungen (Anmeldeinformationen von AWS Secrets Manager) oder anonyme Bindungen.	`sssd`, `sssd-ldap` `sssd-tools` , und verwandte Pakete (siehe die Anforderungen pro Skript)
`configure-cloudwatch-logs.sh`	Konfiguriert den CloudWatch Amazon-Ag enten so, dass er die Lifecycle-Aktionsp rotokolle des Knotens an Amazon CloudWatch Logs weiterleitet.	`amazon-cloudwatch-agent`

> **Important**
>
> Diese Skripts installieren die erforderlichen Pakete nicht. Installieren Sie die erforderlichen Pakete in Ihrem AMI, bevor Sie ein Skript verwenden. Ein Skript schlägt fehl, wenn eine Voraussetzung fehlt. Bei einigen Skripten muss die Instanzrolle des Knotens über bestimmte Berechtigungen verfügen. Sehen Sie sich die Anforderungen für jedes Skript an.

Verweisen Sie auf ein Skript

Referenzieren Sie ein Skript anhand seiner Amazon S3-URI im `scriptLocation` Feld einer Lifecycle-Aktion. Die Skripts werden jeweils AWS-Region unter dem `aws-pcs-node-lifecycle-scripts` Präfix des AWS`aws-pcs-repo-`*`region`* -managed-Buckets gespeichert. Verwenden Sie den Bucket genauso AWS-Region wie Ihren Cluster.

```
s3://aws-pcs-repo-region/aws-pcs-node-lifecycle-scripts/script-vversion.sh
```

Jedes Skript ist mit semantischer Versionierung versioniert. Sie können eine bestimmte Version anheften oder die neueste Version innerhalb einer Hauptversion nachverfolgen:

- Eine bestimmte Version anheften — Verwenden Sie die vollständige semantische Version, z. B. `mount-efs-v1.0.0.sh` Der Inhalt einer bestimmten Version ist unveränderlich. Verwenden Sie eine angeheftete Version für die Produktion.
- Aktuelle Version verfolgen — Verwenden Sie den Alias für die Hauptversion, z. B. `mount-efs-v1-latest.sh` Der Alias verweist auf die neueste Version innerhalb dieser Hauptversion. Kombinieren Sie es damit `scriptCachingPolicyREFRESH_ON_REBOOT`, um Updates beim Neustart zu erhalten.

Jedes Skript hat eine zugehörige SHA-256 Prüfsummendatei am selben Ort mit einem `.sha256` Suffix (z. B.`mount-efs-v1.0.0.sh.sha256`). Lesen Sie den `.sha256` Wert und legen Sie ihn als den des Skripts fest. `checksum` Der Agent überprüft dann die Integrität des Skripts beim Herunterladen. Weitere Informationen finden Sie unter Integrität des Skripts (Prüfsummen).

Verfügbare Skriptversionen

AWS versioniert jedes Skript unabhängig. In den folgenden Tabellen sind die verfügbaren Versionen der einzelnen Skripts aufgeführt. Verwenden Sie eine Version in der Amazon S3-URI des Skripts, wie unter beschriebenVerweisen Sie auf ein Skript.

`mount-efs.sh`

Version	Datum der Veröffentlichung	Versionshinweise
1.0.0	23. Juli 2026	Erstversion.

`mount-fsx-lustre.sh`

Version	Datum der Veröffentlichung	Versionshinweise
1.0.0	23. Juli 2026	Erstversion.

`configure-efs-homes.sh`

Version	Datum der Veröffentlichung	Versionshinweise
1.0.0	23. Juli 2026	Erstversion.

`configure-ldaps.sh`

Version	Datum der Veröffentlichung	Versionshinweise
1.0.0	23. Juli 2026	Erstversion.

`configure-cloudwatch-logs.sh`

Version	Datum der Veröffentlichung	Versionshinweise
1.0.0	23. Juli 2026	Erstversion.

Beispiel: Mounten Sie ein EFS-Dateisystem mit einem verwalteten Skript

Das folgende Beispiel verweist auf das verwaltete `mount-efs.sh` Skript in der `us-west-2` AWS-Region. Sie übergeben Argumente als benannte Flags. Die Argumente des Skripts finden Sie unterSkript-Argumente.

```
aws pcs update-compute-node-group \
```

```
  --cluster-identifier my-cluster \
  --compute-node-group-identifier my-cng \
  --node-lifecycle-actions '{
    "stages": {
      "nodeBootstrapped": [
        {
          "name": "Mount EFS home directory",
          "scriptSource": {
            "scriptLocation": "s3://aws-pcs-repo-us-west-2/aws-pcs-node-lifecycle-
scripts/mount-efs-v1.0.0.sh"
          },
          "arguments": ["--file-system-id", "fs-12345678", "--mount-point", "/shared"],
          "onError": "TERMINATE",
          "executionPolicy": "EVERY_BOOT"
        }
      ]
    }
  }'
```

Die Instanzrolle des Knotens muss über die `s3:GetObject` Berechtigung für das Skriptobjekt verfügen. Knoten in einem privaten Subnetz können den Bucket über einen S3-Gateway-VPC-Endpunkt erreichen. Weitere Informationen finden Sie unter Speicherorte von Skripten.

Skript-Argumente

Jedes Skript verwendet Argumente mit benannten Flaggen. Übergeben Sie jedes Flag und seinen Wert als separate Elemente des `arguments` Arrays (z. B.`["--file-system-id", "fs-12345678"]`). Alle Skripte werden unter den Betriebssystemen AL2, AL2023, RHEL, Ubuntu und Rocky Linux ausgeführt und unterstützen diese. `root`

`mount-efs.sh`

Flagge	Erforderlich	Standard	Beschreibung
`--file-system-id`	Ja	—	EFS-Dateisystem-ID, z. `fs-012345 6789abcdef0` B.
`--mount-point`	Ja	—	Absoluter Pfad, unter dem gemountet werden soll, z. `/mnt/efs` B.

Flagge	Erforderlich	Standard	Beschreibung
`--options`	Nein	`tls,iam`	Comma-separated Optionen zum Einhängen. In der Standardeinstellung sind Verschlüsselung bei der Übertragung und IAM-Authentifizierung aktiviert.

Bei der `iam` Standardoption muss die Instanzrolle des Knotens `elasticfilesystem:ClientMount` und `elasticfilesystem:ClientWrite` zulassen. Das Subnetz des Knotens muss ein EFS-Mount-Ziel haben, dessen Sicherheitsgruppe eingehendes NFS (TCP 2049) vom Knoten zulässt.

`mount-fsx-lustre.sh`

Flagge	Erforderlich	Standard	Beschreibung
`--fsx-dns-name`	Ja	—	FSx für den DNS-Namen des Lustre-Dateisystems.
`--mount-name`	Ja	—	Der Lustre-Mount-Name des Dateisystems (der `MountName` Wert aus der FSx-Konsole).
`--mount-point`	Ja	—	Absoluter Pfad, unter dem gemountet werden soll, wie z. B. `/fsx`
`--options`	Nein	`noatime,flock`	Comma-separated Optionen zum Einhängen. Die Standardeinstellung folgt der Empfehlung von FSx for Lustre.

Das Subnetz des Knotens muss das Dateisystem erreichen, und die Sicherheitsgruppe des Dateisystems muss eingehenden Lustre-Verkehr (TCP 988 und 1018—1023 für Clients, die reservierte Quellports verwenden) vom Knoten zulassen.

`configure-efs-homes.sh`

Flagge	Erforderlich	Standard	Beschreibung
`--efs-id`	Ja	—	EFS-Dateisystem-ID, z. `fs-012345 6789abcdef0` B.
`--home-base`	Nein	`/home`	Absoluter Pfad, unter dem das EFS-Dateisystem gemountet und unter dem Basisverzeichnisse erstellt werden sollen.
`--options`	Nein	`tls,iam`	Comma-separated Optionen zum Einhängen. In der Standardeinstellung sind Verschlüsselung bei der Übertragung und IAM-Authentifizierung aktiviert.

Dieses Skript erfordert `amazon-efs-utils`nfs-utils, und PAM-Unterstützung`mkhomedir`. Installieren Sie auf RHEL-family Knoten `oddjoboddjob-mkhomedir`, und`authselect`. Installieren Sie auf AL2023 `authselect` explizit. Installieren Sie auf Ubuntu`libpam-modules`. IAM- und Netzwerkanforderungen stimmen überein`mount-efs.sh`.

`configure-ldaps.sh`

Flagge	Erforderlich	Beschreibung
`--ldap-uri`	Ja	LDAP-Server-URI. Muss mit `ldap://` oder `ldaps://` beginnen.
`--search-base`	Ja	LDAP-Suchbasis, z. B. `dc=example,dc=com`
`--secret-arn`	Nein	AWS Secrets Manager-ARN des Geheimnisses, das die Bindungsanmeldeinformationen enthält. Wenn Sie dieses Flag weglassen, konfiguriert das Skript SSSD für eine anonyme Bindung.

Flagge	Erforderlich	Beschreibung
`--tls-cacert`	Nein	Absoluter Pfad zu einem CA-Bundle für die TLS-Überprüfung. Standardmäßig wird das CA-Paket des Betriebssystems verwendet.

Dieses Skript erfordert `sssd`sssd-ldap, und`sssd-tools`. Auf RHEL-family Knoten erfordert es auch `authselect` oder `authconfig` und`oddjob-mkhomedir`. Auf Ubuntu erfordert es`libpam-modules`.

Mit `--secret-arn` muss die Instanzrolle des Knotens das Geheimnis zulassen`secretsmanager:GetSecretValue`, und der Knoten benötigt die `python3` Pakete `awscli` und. Der Wert des Geheimnisses muss im JSON-Format vorliegen`{"username": "<bind-dn>", "password": "<bind-password>"}`. Anonymous Bind führt keine AWS API-Aufrufe durch und benötigt keine IAM-Berechtigungen.

`configure-cloudwatch-logs.sh`

Flagge	Erforderlich	Standard	Beschreibung
`--log-group-name`	Nein	`/aws/pcs/${PCS_CLUSTER_ID}/lifecycle`	Die CloudWatch Logs-Protokollgruppe, an die weitergeleitet werden soll. Wenn es weggelassen wird, leitet das Skript den Namen von der `PCS_CLUSTER_ID` Umgebungsvariablen ab, die AWS PCS festlegt.

Das Skript leitet das Verzeichnis für das Lifecycle-Aktionsprotokoll weiter ()`/var/log/amazon/pcs/lifecycle/actions`. Die Instanzrolle des Knotens muss dies zulassen `logs:CreateLogGrouplogs:CreateLogStream`, und für `logs:PutLogEvents` die Zielprotokollgruppe. Verwenden Sie dieses Skript, um Logs für das Post-Mortem-Debugging außerhalb der Instanz weiterzuleiten.

> **Note**
>
> Der CloudWatch Amazon-Agent muss als ausgeführt werden, `root` um diese Protokolle weiterzuleiten. Nur der `root` Benutzer kann lesen`/var/log/amazon/pcs/lifecycle/actions`.

> **Note**
>
> Das Skript schreibt seine Konfiguration in eine dedizierte Fragmentdatei. Der CloudWatch Amazon-Agent kombiniert dieses Fragment mit einer vorhandenen Konfiguration, ohne es zu überschreiben, sodass das Skript Ihre vorhandenen Metriken und Protokollgruppeneinstellungen beibehält.

Bewährte Methoden für Node-Lifecycle-Aktionen in AWS STCK

Folgen Sie diesen bewährten Methoden, wenn Sie Aktionen im Knotenlebenszyklus verwenden.

- Testen Sie Crawl-Walk-Run. Verifizieren Sie Skripts auf einer eigenständigen Amazon EC2-Instance, dann als Lifecycle-Aktion auf einer Compute-Knotengruppe mit einer einzelnen statischen Instance (`minInstanceCount`= `maxInstanceCount` = 1), dann mit dynamischer Skalierung.
- Wählen Sie das Richtige. **`executionPolicy`** Belassen Sie das Einmal-Setup auf der Standardeinstellung`FIRST_BOOT_ONLY`; verwenden Sie es `EVERY_BOOT` nur für Konfigurationen, die nach einem Neustart erneut angewendet werden müssen.
- Macht **`EVERY_BOOT`** Skripte idempotent. Re-running sie sollten sicher sein. `FIRST_BOOT_ONLY`Skripte werden einmal ausgeführt, also brauchen sie das nicht.
- Verwenden Sie beschreibende Namen. Namen erscheinen in Protokollpfaden und in der Konsole — zum Beispiel`Mount EFS home directory`, nicht`script1`.
- Binden Sie umfangreiche Installationen in das AMI ein. Der Knoten hat ungefähr 30 Minuten Zeit, um Slurm zu erreichen. Wenn eine Installation länger als 5 Minuten dauert, speichern Sie sie in einem benutzerdefinierten AMI oder in einem gemeinsam genutzten Dateisystem.
- Verwenden Sie Prüfsummen in der Produktion, insbesondere für Skripts aus gemeinsam genutzten Repositorys.
- Leiten Sie Protokolle CloudWatch mit einem Node-Bootstrap-Skript an Amazon weiter, damit Sie beendete Instances debuggen können. Das `configure-cloudwatch-logs.sh` Skript AWS-

maintained erledigt dies. Weitere Informationen finden Sie unter Verwenden Sie AWS-gepflegte Skripte für Node-Lifecycle-Aktionen in AWS STCK.

- Halten Sie die Abhängigkeiten so gering wie möglich und vermeiden Sie das Forken von Hintergrundprozessen, die das Skript überleben.
- Ändern Sie keine AWS PCS-Agentendateien unter `/etc/amazon/pcs/` oder- oder systemd-Units`/var/log/amazon/pcs/`, deren Namen mit beginnen. `pcs-` Diese verwalten die Cluster-Konnektivität.
- Starten Sie die Instanz nicht während eines Lifecycle-Aktionsskripts neu. Ein Neustart unterbricht die AWS PCS-Bootstrap-Sequenz und `slurmd` verhindert so, dass AWS PCS den Knoten startet oder beendet.
- Speichern Sie Geheimnisse im AWS Secrets Manager oder Systems Manager Parameter Store. Übergeben Sie keine Geheimnisse als Skriptargumente. Argumente sind in den API-Antworten und in der Konsole sichtbar.
- Verwenden Sie S3 mit einem Gateway-VPC-Endpunkt für Knoten in privaten Subnetzen. Der Endpunkt bietet Skriptzugriff, ohne dass ein Internetausgang erforderlich ist.

Problembehandlung bei Aktionen im Knotenlebenszyklus in AWS STCK

Verwenden Sie diese Informationen zur Problembehandlung, um häufig auftretende Probleme mit Aktionen im Knotenlebenszyklus zu lösen.

Symptom	Wahrscheinliche Ursache	Was ist zu tun
Der Knoten wird beim Start beendet	Ein `nodeBoots trapped nodeReady` ODER-Skript wurde ungleich Null mit beendet `onError=TERMINATE`	Überprüfen Sie `/var/log/amazon/pcs/lifecycle/actions/<stage>/<name>.log` . Wird `onError=STOP_SEQUENCE` vorübergehend gesetzt, um den Knoten für das Debuggen beizubehalten.
Der Skriptdownload schlägt fehl	Fehlts`3:GetObject` , kein Netzwerkausgang oder falsche URL	Überprüfen Sie die Instanz-IAM-Rolle und das VPC-Routing oder die Endpunkte. Bestätigen Sie die URL und (für S3). `s3VersionId`

Symptom	Wahrscheinliche Ursache	Was ist zu tun
Prüfsumme stimmt nicht überein	Der Inhalt des Skripts wurde geändert oder ist beschädigt	Generieren Sie die Konfiguration neu SHA-256 und aktualisieren Sie sie, oder entfernen Sie die Prüfsumme während der Iteration.
Der Knoten wird nie bereit	Skripts überschreiten das etwa 30-minütige Registrierungsfenster	Verschieben Sie lange Installationen auf das AMI oder ein gemeinsam genutztes Dateisystem; teilen Sie die Arbeit schrittweise auf.
Das aktualisierte Skript wurde nicht angewendet	`CACHE_ONCE` behält die Originalkopie bei	Verwenden Sie `REFRESH_ON_REBOOT` die Instanz oder ersetzen Sie sie.
Das Skript wurde nach dem Neustart nicht erneut ausgeführt	`executionPolicy` ist `FIRST_BOOT_ONLY` (Standard)	`executionPolicy=EV ERY_BOOT` Für Skripts festgelegt, die bei jedem Start ausgeführt werden müssen.
`sinfo`oder `scontrol` nicht in einem Skript gefunden	Das Skript läuft auf der `nodeBootstrapped` Bühne, bevor `slurmd`	Verschiebe die Slurm-dependent Arbeit auf die `nodeReady` Bühne.
Nach der Kündigung können keine Protokolle mehr angezeigt werden	Die Protokolle sind lokal für die Instanz	Fügen Sie ein Node-Bootstrap-Skr ipt hinzu, das Protokolle an Amazon weiterleitet. CloudWatch

Einskalieren AWS STCK

AWS PCS passt die Rechenkapazität Ihres Clusters automatisch an, sodass Ihre Jobs über die Ressourcen verfügen, die sie benötigen. Die Skalierung hängt von den Jobs ab, die Sie einreichen, und davon, wie Sie Ihre Rechenknotengruppen konfigurieren.

Themen

- Job-level Skalierung

Job-level Skalierung

AWS PCS verwendet Skalierung auf Jobebene, um die dynamische Kapazität Ihres Clusters zu verwalten (die Anzahl der Instanzen zwischen der Mindest- und Höchstanzahl, die Sie für eine Rechenknotengruppe festgelegt haben). In regelmäßigen Abständen überprüft AWS PCS wartende Jobs, startet Amazon EC2-Instances, um sie zu erfüllen, und skaliert die dynamische Kapazität wieder nach unten, wenn Jobs abgeschlossen sind und Instances inaktiv werden. Sie verwalten diesen Prozess nicht. AWS PCS bewertet Ihre Arbeitslast und passt die Kapazität für Sie an.

Die statische Kapazität (das Minimum, das Sie immer aktiv halten müssen) wird durch die Skalierung auf Auftragsebene nicht beeinträchtigt. Weitere Informationen finden Sie unter Erstellen einer Rechenknotengruppe in PCS. AWS

Wie Planung und Skalierung zusammenarbeiten

Wenn Sie einen Job einreichen, wählt Slurm Clusterknoten aus und weist sie zu, die die Anforderungen des Jobs erfüllen. Für zugewiesene Knoten, die durch dynamische Kapazität abgesichert sind, startet AWS PCS die Amazon EC2-Instances, die zur Sicherung dieser Knoten erforderlich sind, und weist jede Instance ihrem Knoten zu. Sie verwalten diese Amazon EC2-Instances nicht direkt. AWS PCS kümmert sich in Ihrem Namen um das Starten und Freigeben dynamischer Kapazitäten.

Wie AWS PCS stellt Kapazität für einen Auftrag zur Verfügung

Ein Job wird erst ausgeführt, wenn alle erforderlichen Knoten verfügbar sind. Er beginnt nicht mit einer teilweisen Zuweisung. Wie AWS PCS Kapazität bereitstellt, hängt von den Anforderungen des Jobs und der Konfiguration Ihrer Rechenknotengruppe ab. Dies hängt beispielsweise davon ab, wie viele Knoten der Auftrag benötigt, ob er mehrere Rechenknotengruppen umfasst, ob mehr

Kapazität angefordert wird, als eine einzelne Amazon EC2-Anfrage bereitstellen kann, und ob die Zielknotengruppe durch eine Kapazitätsreservierung abgesichert ist.

AWS PCS bewertet jeden Job und wählt den geeigneten Ansatz aus. In der Praxis nimmt dies eine von zwei Formen an:

- Alles auf einmal. AWS PCS bietet alle Knoten, die der Job benötigt, zusammen oder keine. Der Job beginnt erst, wenn die volle Kapazität verfügbar ist.
- Schrittweise. Für einige Jobs, z. B. solche, die mehr Kapazität benötigen, als eine einzelne Amazon EC2-Anfrage bereitstellen kann, stellt AWS PCS die Kapazität bereit, sobald sie verfügbar ist. Dabei werden die bereits erworbenen Daten gespeichert, während der gesamte Satz abgearbeitet wird.

Wenn eine Compute-Knotengruppe die On-Demand Kaufoption verwendet und mehr als ein Subnetz auflistet, startet AWS PCS die Instances in der Subnetzreihenfolge, die Sie in der `CreateComputeNodeGroup` OR-Anfrage angegeben haben. `UpdateComputeNodeGroup` Wenn ein Subnetz die angeforderte Kapazität nicht bereitstellen kann, versucht AWS PCS das nächste Subnetz in der Liste und fährt fort, bis verfügbare Kapazität gefunden wird.

Wenn die volle Kapazität eines Jobs nicht verfügbar ist, wird er nicht gestartet. AWS PCS startet möglicherweise keine Instances für den Job, oder es gibt möglicherweise Instanzen frei, die es gestartet hat. Der Job steht noch aus, und AWS PCS bewertet ihn in späteren Intervallen erneut, bis seine Kapazität bereitgestellt werden kann.

Example Umfangreicher Auftrag mit inkrementeller Kapazitätsbereitstellung

Angenommen, Sie reichen einen Auftrag ein, für den eine große Anzahl von Knoten erforderlich ist, also mehr, als gleichzeitig bereitgestellt werden können. AWS PCS startet Instances für einen Teil der Knoten des Jobs, weist sie zu und behält diese Kapazität bei, während in späteren Intervallen die verbleibenden Knoten abgerufen werden. Der Job beginnt, sobald alle erforderlichen Knoten verfügbar sind. Wenn Sie sicherstellen müssen, dass beim Start des Jobs die volle Kapazität verfügbar ist, sichern Sie die Compute-Knotengruppe mit einer Kapazitätsreservierung ab, wie im nächsten Abschnitt beschrieben.

Garantiert die Kapazität für große oder zeitkritische Jobs

Wenn ein Auftrag eine große Menge an Kapazität benötigt oder zu einem bestimmten Zeitpunkt beginnen muss, ist die Kapazität auf Abruf möglicherweise nicht verfügbar, wenn Sie ihn einreichen.

Um Kapazität im Voraus zu reservieren, damit sie verfügbar ist, wenn Ihr Job ausgeführt wird, können Sie On-Demand Kapazitätsreservierungen (ODCRs) oder Amazon EC2 Capacity Blocks for ML verwenden.

Konfigurieren Sie Ihre Rechenknotengruppe so, dass sie die Reservierung verwendet, und senden Sie dann Jobs an eine Warteschlange, die dieser Knotengruppe zugeordnet ist. Reservierte Kapazität wird gemäß Ihrer ODCR oder Ihrem Kapazitätsblock in Rechnung gestellt, unabhängig davon, ob ein Job für sie ausgeführt wird oder nicht. Weitere Informationen finden Sie unter Verwenden von ODCRs mit AWS PCS und Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit PCS. AWS

Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS

In Amazon EC2 kann eine Startvorlage eine Reihe von Einstellungen speichern, sodass Sie diese beim Starten von Instances nicht einzeln angeben müssen. AWS PCS enthält Startvorlagen als flexible Methode zur Konfiguration von Rechenknotengruppen. Wenn Sie eine Knotengruppe erstellen, stellen Sie eine Startvorlage bereit. AWS PCS erstellt daraus eine abgeleitete Startvorlage, die Transformationen enthält, um sicherzustellen, dass sie mit dem Service funktioniert.

Wenn Sie wissen, welche Optionen und Überlegungen beim Schreiben einer benutzerdefinierten Startvorlage zu beachten sind, können Sie eine Vorlage für die Verwendung mit AWS PCS erstellen. Weitere Informationen zu Startvorlagen finden Sie unter Launching an Instance from a Launch an Instance from a Launch Template im Amazon EC2 EC2-Benutzerhandbuch.

Themen

- Überblick über Startvorlagen in AWS PCS
- Erstellen einer grundlegenden Startvorlage
- Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK
- Kapazitätsreservierungen in AWS 5 STÜCK
- Nützliche Parameter für Startvorlagen

Überblick über Startvorlagen in AWS PCS

Es stehen über 30 Parameter zur Verfügung, die Sie in eine EC2-Startvorlage aufnehmen können und die viele Aspekte der Konfiguration von Instances steuern. Die meisten sind vollständig mit AWS PCS kompatibel, es gibt jedoch einige Ausnahmen.

Die folgenden Parameter der EC2 Launch-Vorlage werden von AWS PCS ignoriert, da diese Eigenschaften direkt vom Dienst verwaltet werden müssen:

- Attribute des type/Specify Instance-Instance-Typs (`InstanceRequirements`) — AWS PCS unterstützt keine attributbasierte Instance-Auswahl.
- Instanztyp (`InstanceType`) — Geben Sie Instanztypen an, wenn Sie eine Knotengruppe erstellen.

- Erweitertes details/IAM Instanzprofil (`IamInstanceProfile`) — Dieses geben Sie an, wenn Sie die Knotengruppe erstellen oder aktualisieren.
- Erweiterte details/Disable API-Terminierung (`DisableApiTermination`) — AWS PCS muss den Lebenszyklus der von ihm gestarteten Knotengruppen-Instances kontrollieren.
- Erweiterter details/Disable API-Stopp (`DisableApiStop`) — AWS PCS muss den Lebenszyklus der von ihm gestarteten Knotengruppen-Instances kontrollieren.
- Erweitert details/Stop — Verhalten im Ruhezustand (`HibernationOptions`) — AWS PCS unterstützt den Ruhezustand von Instanzen nicht.
- Advanced details/Elastic GPU (`ElasticGpuSpecifications`) — Amazon Elastic Graphics hat am 8. Januar 2024 das Ende der Nutzungsdauer erreicht.
- Erweiterte details/Elastic Inferenz (`ElasticInferenceAccelerators`) — Amazon Elastic Inference ist für Neukunden nicht mehr verfügbar.
- AAdvanced details/Specify CPU options/Threadspro Kern (`ThreadsPerCore`) — AWS PCS legt die Anzahl der Threads pro Kern auf 1 fest.

Für diese Parameter gelten spezielle Anforderungen, die die Kompatibilität mit AWS PCS unterstützen:

- Benutzerdaten (`UserData`) — Diese müssen mehrteilig codiert sein. Siehe Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK.
- Anwendungs- und Betriebssystem-Images (`ImageId`) — Sie können dies einschließen. Wenn Sie jedoch beim Erstellen oder Aktualisieren der Knotengruppe eine AMI-ID angeben, überschreibt diese den Wert in der Startvorlage. Das von Ihnen bereitgestellte AMI muss mit AWS PCS kompatibel sein. Weitere Informationen finden Sie unter "Amazon Machine Images (AMIs) für AWS STK..
- Netzwerk settings/Firewall (Sicherheitsgruppen) (`SecurityGroups`) — Eine Liste von Sicherheitsgruppennamen kann in einer AWS PCS-Startvorlage nicht festgelegt werden. Sie können eine Liste von Sicherheitsgruppen IDs (`SecurityGroupIds`) einrichten, es sei denn, Sie definieren Netzwerkschnittstellen in der Startvorlage. Anschließend müssen Sie IDs für jede Schnittstelle eine Sicherheitsgruppe angeben. Weitere Informationen finden Sie unter Sicherheitsgruppen in AWS STK..
- settings/Advanced Netzwerkkonfiguration (`NetworkInterfaces`) — Wenn Sie EC2-Instances mit einer einzigen Netzwerkkarte verwenden und keine spezielle Netzwerkkonfiguration benötigen, kann AWS PCS das Instance-Netzwerk für Sie konfigurieren. Um mehrere Netzwerkkarten zu konfigurieren oder den Elastic Fabric Adapter auf Ihren Instances zu aktivieren, verwenden

SieNetworkInterfaces. IDs Unter jeder Netzwerkschnittstelle muss eine Liste der Sicherheitsgruppen enthalten seinGroups. Weitere Informationen finden Sie unter Mehrere Netzwerkschnittstellen in AWS PCS.

- Erweiterte Details/Kapazitätsreservierung (CapacityReservationSpecification) — Dies kann eingestellt werden, kann aber CapacityReservationId bei der Arbeit mit AWS PCS nicht auf ein bestimmtes Objekt verweisen. Sie können jedoch auf eine Kapazitätsreservierungsgruppe verweisen, wenn diese Gruppe eine oder mehrere Kapazitätsreservierungen enthält. Weitere Informationen finden Sie unter Kapazitätsreservierungen in AWS 5 STÜCK.

Erstellen einer grundlegenden Startvorlage

Sie können eine Startvorlage mit dem AWS-Managementkonsole oder dem erstellen AWS CLI.

AWS-Managementkonsole

Eine Startvorlage erstellen

1. Öffnen Sie die EC2Amazon-Konsole und wählen Sie Vorlagen starten aus.
2. Wählen Sie Startvorlage erstellen.
3. Geben Sie unter Name und Beschreibung der Startvorlage einen eindeutigen, unverwechselbaren Namen für den Namen der Startvorlage ein
4. Wählen Sie unter key pair (Anmeldung) bei Schlüsselpaarname das SSH-Schlüsselpaar aus, das für die Anmeldung bei von AWS PCS verwalteten EC2 Instanzen verwendet werden soll. Dies ist zwar optional, wird aber empfohlen.
5. Wählen Sie unter Netzwerkeinstellungen und dann Firewall (Sicherheitsgruppen) die Sicherheitsgruppen aus, die an die Netzwerkschnittstelle angehängt werden sollen. Alle Sicherheitsgruppen in der Startvorlage müssen aus Ihrer AWS PCS-Cluster-VPC stammen. Wählen Sie mindestens:

   - Eine Sicherheitsgruppe, die die Kommunikation mit dem AWS PCS-Cluster ermöglicht
   - Eine Sicherheitsgruppe, die die Kommunikation zwischen EC2 Instances ermöglicht, die von AWS PCS gestartet wurden
   - (Optional) Eine Sicherheitsgruppe, die eingehenden SSH-Zugriff auf interaktive Instanzen ermöglicht
   - (Optional) Eine Sicherheitsgruppe, die es Rechenknoten ermöglicht, ausgehende Verbindungen zum Internet herzustellen

- (Optional) Sicherheitsgruppe (n), die den Zugriff auf Netzwerkressourcen wie gemeinsam genutzte Dateisysteme oder einen Datenbankserver ermöglichen.

6. Ihre neue Startvorlagen-ID ist in der EC2 Amazon-Konsole unter Startvorlagen verfügbar. Die ID der Startvorlage wird das folgende Formular haben`lt-0123456789abcdef01`.

Als nächster Schritt wird empfohlen

- Verwenden Sie die neue Startvorlage, um eine AWS PCS-Compute-Knotengruppe zu erstellen oder zu aktualisieren.

AWS CLI

Eine Startvorlage erstellen

Erstellen Sie Ihre Startvorlage mit dem folgenden Befehl.

- Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:

  a. *region-code*Ersetzen Sie es durch die AWS-Region Stelle, an der Sie mit AWS PCS arbeiten

  b. *my-launch-template-name*Ersetzen Sie es durch einen Namen für Ihre Vorlage. Es muss für das AWS-Konto und, das AWS-Region Sie verwenden, eindeutig sein.

  c. *my-ssh-key-name*Ersetzen Sie es durch den Namen Ihres bevorzugten SSH-Schlüssels.

  d. Ersetzen Sie *sg-ExampleID1* und *sg-ExampleID2* durch eine Sicherheitsgruppe IDs , die die Kommunikation zwischen Ihren EC2 Instances und dem Scheduler sowie die Kommunikation zwischen EC2 Instanzen ermöglicht. Wenn Sie nur über eine Sicherheitsgruppe verfügen, die den gesamten Datenverkehr ermöglicht, können Sie das vorangegangene Kommazeichen entfernen`sg-ExampleID2`. Sie können auch weitere Sicherheitsgruppen IDs hinzufügen. Alle Sicherheitsgruppen, die Sie in die Startvorlage aufnehmen, müssen aus Ihrer AWS PCS-Cluster-VPC stammen.

```
aws ec2 create-launch-template --region region-code \
    --launch-template-name my-template-name \
    --launch-template-data '{"KeyName":"my-ssh-key-name","SecurityGroupIds":
 ["sg-ExampleID1","sg-ExampleID2"]}'
```

Es AWS CLI wird Text ausgegeben, der dem folgenden ähnelt. Die ID der Startvorlage befindet sich in. `LaunchTemplateId`

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0123456789abcdef01",
        "LaunchTemplateName": "my-launch-template-name",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-04-30T18:16:06.000Z"
    }
}
```

Als nächster Schritt wird empfohlen

- Verwenden Sie die neue Startvorlage, um eine AWS PCS-Compute-Knotengruppe zu erstellen oder zu aktualisieren.

Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK

> **Note**
>
> Für die meisten Knotenkonfigurationsaufgaben empfehlen wir Aktionen im Knotenlebenszyklus. Sie bieten eine wiederverwendbare API-native Konfiguration mit Fehlerbehandlung pro Skript und Neustartsteuerung. Verwenden Sie die Benutzerdaten der Startvorlage für Aufgaben, die ausgeführt werden müssen, bevor der AWS PCS-Agent gestartet wird. Weitere Informationen finden Sie unter Aktionen im Lebenszyklus eines Knotens.

Sie können EC2-Benutzerdaten in Ihrer Startvorlage angeben, die beim Start Ihrer Instances `cloud-init` ausgeführt wird. Benutzerdatenblöcke mit dem Inhaltstyp werden `cloud-config` ausgeführt, bevor sich die Instance bei der AWS PCS-API registriert, während Benutzerdatenblöcke mit dem Inhaltstyp nach Abschluss der Registrierung, aber bevor der Slurm-Daemon gestartet wird, `text/x-shellscript` ausgeführt werden. Weitere Informationen zu Inhaltstypen finden Sie in der Cloud-Init-Dokumentation.

Unsere Benutzerdaten können gängige Konfigurationsszenarien ausführen, einschließlich, aber nicht beschränkt auf die folgenden:

- Einschließlich Benutzer oder Gruppen
- Pakete installieren
- Partitionen und Dateisysteme erstellen
- Mounten von Netzwerkdateisystemen

Benutzerdaten in Startvorlagen müssen im mehrteiligen https://cloudinit.readthedocs.io/en/latest/topics/format.html#mime-multi-part-archive MIME-Archivformat vorliegen. Dies liegt daran, dass Ihre Benutzerdaten mit anderen AWS PCS-Benutzerdaten zusammengeführt werden, die für die Konfiguration von Knoten in Ihrer Knotengruppe erforderlich sind. Sie können mehrere Benutzerdatenblöcke in einer einzelnen mehrteiligen MIME-Datei kombinieren.

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 Linie vor dieser Grenze leer lassen.
  - Die Inhaltstypdeklaration für den Block: `Content-Type: text/cloud-config; charset="us-ascii"` oder`Content-Type: text/x-shellscript; charset="us-ascii"`. Sie müssen die Zeile nach der Inhaltstypdeklaration leer lassen.
  - Der Inhalt der Benutzerdaten, z. B. eine Liste von Shell-Befehlen oder `cloud-config` -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 Amazon EC2-Konsole hinzufügen, können Sie sie als Klartext einfügen. Sie können es auch aus einer Datei hochladen. Wenn Sie das AWS CLI oder ein AWS SDK verwenden, müssen Sie zuerst die Benutzerdaten mit

Base64 kodieren und diese Zeichenfolge beim Aufruf als Wert des `UserData` Parameters einreichen CreateLaunchTemplate, wie in dieser JSON-Datei gezeigt.

```
{
    "LaunchTemplateName": "base64-user-data",
    "LaunchTemplateData": {
        "UserData":
 "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
    }
}
```

Beispiele

- Beispiel: Software aus einem Paket-Repository installieren
- Beispiel: Führen Sie Skripts aus einem S3-Bucket aus
- Beispiel: Legen Sie globale Umgebungsvariablen fest
- Beispiel: Führen Sie frühe Startvorgänge aus
- Verwenden von Netzwerkdateisystemen mit AWS PCS
- Beispiel: Verwenden Sie ein EFS-Dateisystem als gemeinsam genutztes Home-Verzeichnis

Beispiel: Software für AWS PCS aus einem Paket-Repository installieren

Geben Sie dieses Skript als Wert von `"userData"` in Ihrer Startvorlage an. Weitere Informationen finden Sie unter Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK.

Dieses Skript verwendet cloud-config, um beim Start Softwarepakete auf Knotengruppen-Instances zu installieren. Weitere Informationen finden Sie unter Benutzerdatenformate in der Cloud-Init-Dokumentation. In diesem Beispiel wird und installiert`curl`. `llvm`

> **Note**
>
> Ihre Instances müssen in der Lage sein, eine Verbindung zu ihren konfigurierten Paket-Repositorys herzustellen.

```
MIME-Version: 1.0
```

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

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

packages:
- python3-devel
- rust
- golang

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

Beispiel: Zusätzliche Skripts für AWS PCS aus einem S3-Bucket ausführen

Geben Sie dieses Skript als Wert von `"userData"` in Ihrer Startvorlage an. Weitere Informationen finden Sie unter Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK.

Das folgende Benutzerdatenskript verwendet cloud-config, um ein Skript aus einem S3-Bucket zu importieren und es beim Start auf Knotengruppen-Instances auszuführen. Weitere Informationen finden Sie unter Benutzerdatenformate in der Cloud-Init-Dokumentation.

Ersetzen Sie die folgenden Werte durch Ihre eigenen Daten:

- *`amzn-s3-demo-bucket`*— Der Name eines S3-Buckets, aus dem Ihr Konto lesen kann.
- *`object-key`*— Der S3-Objektschlüssel des zu importierenden Skripts. Dazu gehören der Name des Skripts und sein Speicherort in der Ordnerstruktur des Buckets. Beispiel, `scripts/script.sh`. Weitere Informationen finden Sie unter Organisieren von Objekten in der Amazon S3 S3-Konsole mithilfe von Ordnern im Amazon Simple Storage Service-Benutzerhandbuch.
- *`shell`*— Die Linux-Shell, die zur Ausführung des Skripts verwendet werden soll, z. `bash` B.

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

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

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/object-key /tmp/script.sh
- /usr/bin/shell /tmp/script.sh
```

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

Das IAM-Instanzprofil für die Knotengruppe muss Zugriff auf den Bucket haben. Die folgende IAM-Richtlinie ist ein Beispiel für den Bucket im obigen Benutzerdatenskript.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Beispiel: Legen Sie globale Umgebungsvariablen für AWS PCS fest

Geben Sie dieses Skript als Wert von `"userData"` in Ihrer Startvorlage an. Weitere Informationen finden Sie unter Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK.

Im folgenden Beispiel werden globale Variablen `/etc/profile.d` für Knotengruppen-Instances festgelegt.

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

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

#!/bin/bash
touch /etc/profile.d/awspcs-userdata-vars.sh
```

```
echo MY_GLOBAL_VAR1=100 >> /etc/profile.d/awspcs-userdata-vars.sh
echo MY_GLOBAL_VAR2=abc >> /etc/profile.d/awspcs-userdata-vars.sh

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

Beispiel: Führen Sie frühe Startvorgänge mit einem Cloud-Init-Boothook aus

Geben Sie dieses Skript als Wert von `"userData"` in Ihrer Startvorlage an. Weitere Informationen finden Sie unter Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK.

Einige Konfigurationen müssen ganz am Anfang der Startsequenz ausgeführt werden, bevor der AWS PCS-Bootstrap gestartet wird und bevor ein Dienst die Hardware verwendet. Typische Fälle sind GPU-Konfigurationsbefehle, die nicht wirksam werden können, während Prozesse die GPUs halten, wie z. B. das Aktivieren des Multi-Instance NVIDIA-GPU-Modus (MIG), und Vorgänge, für deren Wirksamkeit ein Neustart erforderlich ist, z. B. das Ändern des GPU-ECC-Modus. Ein Neustart in dieser Phase ist sicher, da der AWS PCS-Bootstrap noch nicht gestartet wurde. Führen Sie keinen Neustart zu einem späteren Zeitpunkt des Startvorgangs durch, z. B. über ein Aktionsskript für den Node-Lebenszyklus: Ein Neustart dort unterbricht die AWS PCS-Bootstrap-Sequenz. Weitere Informationen finden Sie unter Bewährte Methoden für Node-Lifecycle-Aktionen in AWS STCK.

Um den Boot-Hook zu erstellen, fügen Sie Ihren Benutzerdaten einen Block mit einem Inhaltstyp `text/cloud-boothook` hinzu. Der Block enthält das Skript selbst. Ein Boothook ist der früheste Hook, der für Benutzerdaten verfügbar ist: Er `cloud-init` wird ausgeführt, sobald der Block verarbeitet wird, vor `cloud-config` Anweisungen wie `bootcmd` und vor den AWS PCS-Benutzerdatenblöcken, die den Knoten registrieren. Ein Boothook wird bei jedem Systemstart ausgeführt, einschließlich des Neustarts, den er möglicherweise selbst auslöst. Daher muss das Skript idempotent sein: Prüfen Sie zuerst den aktuellen Status und überspringen Sie bereits ausgeführte Arbeiten. Weitere Informationen dazu finden Sie in der cloud-init-Dokumentation.

Das folgende Beispiel aktiviert den MIG-Modus auf allen GPUs und erstellt die MIG-Instanzen bei jedem Start. Da noch kein GPU-Client gestartet wurde, wird die Modusänderung sofort wirksam und es ist kein Neustart erforderlich.

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

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

#!/bin/bash
```

```
if ! nvidia-smi -i 0 -q | grep -A1 'MIG Mode' | grep -q 'Current.*Enabled'; then
    nvidia-smi -mig 1
fi
if [ -z "$(nvidia-smi -L | grep MIG)" ]; then
    for i in 0 1 2 3 4 5 6 7; do
        nvidia-smi mig -i $i -cgi 9,14,19,19 -C
    done
fi

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

Wenn Ihr Vorgang ausgibt, dass ein Neustart erforderlich ist, starten Sie den Computer nach der Statusänderung vom selben Boothook aus neu. Beim nächsten Systemstart stellen die Idempotenz-Prüfungen fest, dass die Arbeit bereits erledigt ist, sodass der Knoten normal in den AWS PCS-Bootstrap übergeht.

Beispiel: Verwenden Sie ein EFS-Dateisystem als gemeinsam genutztes Home-Verzeichnis für AWS PCS

Geben Sie dieses Skript als Wert für `"userData"` in Ihrer Startvorlage an. Weitere Informationen finden Sie unter Arbeiten mit Amazon EC2-Benutzerdaten für AWS STCK.

In diesem Beispiel wird das EFS-Mount-In zum Beispiel erweitertVerwenden von Netzwerkdateisystemen mit AWS PCS, um ein gemeinsam genutztes Home-Verzeichnis zu implementieren. Der Inhalt von /home wird gesichert, bevor das EFS-Dateisystem bereitgestellt wird. Die Inhalte werden dann nach Abschluss des Mounts schnell an ihren Platz auf dem gemeinsam genutzten Speicher kopiert.

Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *`/mount-point-directory`*— Der Pfad auf einer Instanz, auf der Sie das EFS-Dateisystem mounten möchten.
- *`filesystem-id`*— Die Dateisystem-ID für das EFS-Dateisystem.

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

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

```
packages:
  - amazon-efs-utils

runcmd:
  - mkdir -p /tmp/home
  - rsync -a /home/ /tmp/home
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults
  - rsync -a --ignore-existing /tmp/home/ /home
  - rm -rf /tmp/home/

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

Beispiel: Passwortloses SSH aktivieren

Sie können auf dem Beispiel für ein gemeinsam genutztes Home-Verzeichnis aufbauen, um SSH-Verbindungen zwischen Clusterinstanzen mithilfe von SSH-Schlüsseln zu implementieren. Führen Sie für jeden Benutzer, der das Shared Home-Dateisystem verwendet, ein Skript aus, das dem folgenden ähnelt:

```
#!/bin/bash

mkdir -p $HOME/.ssh && chmod 700 $HOME/.ssh
touch $HOME/.ssh/authorized_keys
chmod 600 $HOME/.ssh/authorized_keys

if [ ! -f "$HOME/.ssh/id_rsa" ]; then
    ssh-keygen -t rsa -b 4096 -f $HOME/.ssh/id_rsa -N ""
    cat ~/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
fi
```

> **Note**
>
> Die Instanzen müssen eine Sicherheitsgruppe verwenden, die SSH-Verbindungen zwischen Clusterknoten ermöglicht.

Kapazitätsreservierungen in AWS 5 STÜCK

Sie können Amazon EC2-Kapazität in einer bestimmten Availability Zone und für eine bestimmte Dauer reservieren, indem Sie On-Demand Capacity Reservations oder Amazon EC2 Capacity Blocks

for ML verwenden, um sicherzustellen, dass Sie die erforderliche Rechenkapazität zur Verfügung haben, wenn Sie sie benötigen.

On-Demand Mit Kapazitätsreservierungen (ODCRs) können Sie Rechenkapazität für Ihre Amazon EC2-Instances in einer bestimmten Availability Zone für einen beliebigen Zeitraum reservieren. Sie können Reservierungen jederzeit erstellen und stornieren, ohne langfristige Verpflichtungen oder Vorauszahlungen. ODCRs sind ideal, wenn Sie flexible Kapazitätsreservierungen benötigen, die Sie ändern können, wenn sich Ihre Anforderungen ändern. Weitere Informationen finden Sie unter On-Demand Kapazitätsreservierungen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Amazon EC2 Capacity Blocks for ML ermöglichen es Ihnen, Instances für GPU-based beschleunigte Datenverarbeitung zu einem späteren Zeitpunkt zu reservieren, um Ihre Workloads für maschinelles Lernen (ML) zu unterstützen. Weitere Informationen finden Sie unter Capacity Blocks for ML im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Themen

- Verwendung ODCRs mit AWS PCS
- Verwenden I-ODCRs mit AWS STK.
- Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit AWS 2 STÜCK

Verwendung ODCRs mit AWS PCS

Sie können wählen, wie AWS PCS Ihre Reserved Instances nutzt. Wenn Sie ein offenes ODCR erstellen, werden alle passenden Instances, die von AWS PCS oder anderen Prozessen in Ihrem Konto gestartet wurden, auf die Reservierung angerechnet. Bei einem gezielten ODCR werden nur Instances, die mit der spezifischen Reservierungs-ID gestartet wurden, auf die Reservierung angerechnet. Bei zeitkritischen Workloads ODCRs sind gezielte Workloads üblicher.

Sie können eine AWS PCS-Compute-Knotengruppe so konfigurieren, dass sie ein zielgerichtetes ODCR verwendet, indem Sie es zu einer Startvorlage hinzufügen. Gehen Sie dazu wie folgt vor:

1. Erstellen Sie mithilfe des Benutzerleitfadens Amazon EC2 Create a Capacity Reservation User Guide eine gezielte On-Demand-Kapazitätsreservierung (ODCR).
2. Ordnen Sie das ODCR einer Startvorlage zu. Es gibt zwei Möglichkeiten, dies zu tun:

    a. Direkte ODCR-Zuordnung: Verweisen Sie direkt in der Startvorlage auf die ODCR-ID. Dieser Ansatz bietet eine strikte Kapazitätskontrolle und unterstützt kein Instanz-Backfilling (wenn

die Compute-Knotengruppe mehr Instanzen anfordert, als im ODCR verfügbar sind, werden keine zusätzlichen Instanzen gestartet).

b. Zuordnung der Kapazitätsreservierungsgruppe: Fügen Sie das ODCR einer Kapazitätsreservierungsgruppe hinzu und verweisen Sie in der Startvorlage auf die Gruppe. Dieser Ansatz unterstützt das Auffüllen von Instanzen, sodass AWS PCS zusätzliche On-Demand-Instances starten kann, wenn die Reservierungskapazität überschritten wird.

3. Erstellen oder aktualisieren Sie eine AWS PCS-Compute-Knotengruppe, um die Startvorlage zu verwenden. Weitere Informationen finden Sie im AWS PCS Compute Node Groups-Benutzerhandbuch.

   - Stellen Sie den `purchaseOption` Wert der Compute-Knotengruppe auf ein`ONDEMAND`.

Beispiel: Reservieren und verwenden Sie hpc6a.48xlarge-Instances mit einem gezielten ODCR

Dieser Beispielbefehl erstellt ein Ziel-ODCR für 32 hpc6a.48xlarge-Instances. Um die Reserved Instances in einer Platzierungsgruppe zu starten, fügen Sie dem Befehl etwas hinzu. `--placement-group-arn` Sie können mit `--end-date` und ein Enddatum definieren`--end-date-type`, andernfalls wird die Reservierung so lange fortgesetzt, bis sie manuell beendet wird.

```
aws ec2 create-capacity-reservation \
    --instance-type hpc6a.48xlarge \
    --instance-platform Linux/UNIX \
    --availability-zone us-east-2a \
    --instance-count 32 \
    --instance-match-criteria targeted
```

Das Ergebnis dieses Befehls ist ein ARN für das neue ODCR. Die ODCR-ID kann aus dem ARN `"arn:aws:ec2:us-east-2:123456789012:capacity-reservation/ODCR-ID"` oder mithilfe von Amazon DescribeCapacityReservations EC2 abgerufen werden.

Direkte ODCR-Zuordnung: Fügen Sie die ODCR-ID zur Startvorlage hinzu. Hier ist ein Beispiel für eine Startvorlage, die auf die ODCR-ID verweist.

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationTarget": {
      "CapacityReservationId": "cr-1234567890abcdef1"
    }
```

```
  }
}
```

Zuordnung von Kapazitätsreservierungsgruppen: Erstellen Sie eine Kapazitätsreservierungsgruppe und fügen Sie die Gruppe zur Startvorlage hinzu. Mit dem folgenden Befehl wird eine Kapazitätsreservierungsgruppe mit dem Namen erstelltEXAMPLE-CR-GROUP.

```
aws resource-groups create-group \
    --name EXAMPLE-CR-GROUP \
    --configuration \
        '{"Type": "AWS::EC2::CapacityReservationPool"}' \
        '{"Type": "AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```

Mit dem folgenden Befehl wird das ODCR zur Gruppe „Kapazitätsreservierung“ hinzugefügt.

```
aws resource-groups group-resources --group EXAMPLE-CR-GROUP \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/
cr-1234567890abcdef1
```

Nachdem das ODCR erstellt und zu einer Kapazitätsreservierungsgruppe hinzugefügt wurde, kann es nun mit einer AWS PCS-Compute-Knotengruppe verbunden werden, indem es zu einer Startvorlage hinzugefügt wird. Hier ist ein Beispiel für eine Startvorlage, die auf die Kapazitätsreservierungsgruppe verweist.

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:us-
east-2:123456789012:group/EXAMPLE-CR-GROUP"
  }
}
```

Erstellen oder aktualisieren Sie abschließend eine AWS PCS-Compute-Knotengruppe, um hpc6a.48xlarge-Instances zu verwenden, und verwenden Sie die Startvorlage, die auf die ODCR verweist. Legen Sie für eine statische Knotengruppe die Mindest- und Höchstzahl der Instanzen auf die Größe der Reservierung fest (32). Legen Sie für eine dynamische Knotengruppe die Mindestanzahl der Instanzen auf 0 und die Höchstzahl auf die gewünschte Instanzgröße fest.

Dieses Beispiel ist eine einfache Implementierung eines einzelnen ODCR, das für eine Rechenknotengruppe bereitgestellt wird. AWS PCS unterstützt jedoch viele andere Designs.

Sie können beispielsweise eine große ODCR- oder Kapazitätsreservierungsgruppe auf mehrere Rechenknotengruppen aufteilen. Oder Sie können ODCRs das verwenden, das ein anderes AWS-Konto erstellt und mit Ihrem geteilt wurde.

Weitere Informationen finden Sie unter On-Demand-Kapazitätsreservierungen und Kapazitätsblöcke für ML im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Verwenden I-ODCRs mit AWS STK.

Mithilfe von Reservierungen für unterbrechbare On-Demand Kapazitäten (I-ODCRs) können ODCR-Besitzer ungenutzte reservierte Kapazität vorübergehend mit anderen Konten in ihrer AWS Organisation teilen. Consumer-Instances erhalten eine 2-minütige Kündigungswarnung, wenn der Eigentümer Kapazität zurückfordert, sodass sie für fehlertolerante Workloads wie Stapelverarbeitung, ML-Training und Datenanalyse I-ODCRs geeignet sind.

Weitere Informationen zu I-ODCRs finden Sie unter Unterbrechungskapazitätsreservierungen im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Wie I-ODCRs arbeiten Sie mit AWS STK.

Ein I-ODCR wird aus einem vorhandenen Quell-ODCR erstellt. Der Besitzer gibt an, wie viele Instanzen der unterbrechbaren Reservierung zugewiesen werden sollen. Diese Instanzen werden vom Quell-ODCR auf das neue übertragen. I-ODCR Der Eigentümer kann jederzeit Kapazität zurückfordern, wodurch Consumer-Instances mit einer Kündigungsfrist von 2 Minuten beendet werden.

Wichtigste Merkmale:

- I-ODCRs sind standardmäßig adressiert — Verbraucher müssen in ihrer Startkonfiguration auf die Reservierungs-ID verweisen.
- I-ODCRs kann nicht zu Kapazitätsreservierungsgruppen hinzugefügt werden.
- Pro Quell-ODCR kann nur eine unterbrechbare Zuordnung erstellt werden.
- Wenn der Eigentümer Kapazität zurückfordert, gibt es keinen Fallback auf On-Demand oder Spot-Consumer-Instances.

Konfiguration eines AWS PCS-Compute-Knotengruppe zur Verwendung einer I-ODCR

Sie können eine AWS PCS-Compute-Knotengruppe so konfigurieren, dass sie eine gemeinsam genutzte Knotengruppe verwendet, I-ODCR indem Sie sie zu einer Startvorlage hinzufügen. Hier sind die Schritte:

- Stellen Sie sicher, dass Sie Zugriff auf die haben I-ODCR. Der ODCR-Besitzer muss die unterbrechbare Reservierung mithilfe von AWS Resource Access Manager (RAM) mit Ihrem Konto teilen. Sobald es geteilt wurde, I-ODCR erscheint es in Ihrem Konto unter Kapazitätsreservierungen auf der Amazon EC2 EC2-Konsole.
- Erstellen Sie eine Startvorlage, die auf die I-ODCR abzielt. Verweisen Sie direkt auf die I-ODCR ID und legen Sie den Markttyp auf fest`interruptible-capacity-reservation`. Hier ist ein Beispiel für eine Startvorlage:

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationTarget": {
      "CapacityReservationId": "cr-1234567890abcdef1"
    }
  },
  "InstanceMarketOptions": {
    "MarketType": "interruptible-capacity-reservation"
  }
}
```

- Erstellen oder aktualisieren Sie eine AWS PCS-Compute-Knotengruppe, um die Startvorlage zu verwenden. Weitere Informationen finden Sie unter AWS PCS-Rechenknotengruppen.
  - Stellen Sie die `purchaseOption` der Compute-Knotengruppe auf ein`INTERRUPTIBLE_CAPACITY_RESERVATION`.

Umgang mit Unterbrechungen

Wenn der I-ODCR Eigentümer Kapazität zurückfordert, erhalten Consumer-Instances von Amazon eine zweiminütige Kündigungswarnung. EventBridge Gehen Sie wie folgt vor, um Unterbrechungen Ihrer PCS-Workloads ordnungsgemäß zu handhaben: AWS

- Konfigurieren Sie Ihre Anwendungen so, dass sie auf Unterbrechungsereignisse warten. EventBridge

- Implementieren Sie Checkpointing, damit Jobs Zwischenergebnisse speichern und nach einer Unterbrechung wieder aufgenommen werden können.
- Legen Sie für Compute-Knotengruppen mit einer dynamischen Skalierungskonfiguration die Mindestanzahl der Instanzen auf fest, `0` sodass die Gruppe problemlos herunterskaliert werden kann, wenn Kapazität zurückgewonnen wird.

Weitere Informationen zur Überwachung von Unterbrechungsereignissen finden Sie unter Überwachen von unterbrechbaren Kapazitätsreservierungen mit EventBridge im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Beispiel: Teilen und verwenden Sie hpc7a.96xlarge-Instances mit einem I-ODCR

In diesem Beispiel wird beschrieben, wie I-ODCR aus einem vorhandenen ODCR eine erstellt und mit einer PCS-Compute-Knotengruppe verwendet wird. AWS

Schritt 1: Erstellen Sie die unterbrechbare Reservierung aus einem Quell-ODCR.

Der ODCR-Besitzer erstellt aus seiner bestehenden Reservierung mit 32 Instanzen eine unterbrechbare Zuteilung von 16 Instanzen:

```
aws ec2 create-interruptible-capacity-reservation-allocation \
    --capacity-reservation-id cr-source1234567890a \
    --instance-count 16
```

Das Quell-ODCR zeigt jetzt 16 Instanzen an, und es I-ODCR wird eine neue mit 16 Instanzen erstellt.

Schritt 2: Teilen Sie das I-ODCR verwendende AWS RAM.

Der Eigentümer teilt das I-ODCR mit dem Verbraucherkonto:

```
aws ram create-resource-share \
    --name "HPC-Interruptible-Share" \
    --resource-arns arn:aws:ec2:us-east-2:123456789012:capacity-reservation/cr-
interruptible456 \
    --principals 987654321098
```

Schritt 3: Erstellen Sie eine Startvorlage für die I-ODCR.

Der Verbraucher erstellt eine Vorlage für die Markteinführung:

```
{
  "CapacityReservationSpecification": {
    "CapacityReservationTarget": {
      "CapacityReservationId": "cr-interruptible456"
    }
  },
  "InstanceMarketOptions": {
    "MarketType": "interruptible-capacity-reservation"
  }
}
```

Schritt 4: Erstellen Sie mithilfe der Startvorlage eine AWS PCS-Compute-Knotengruppe.

Erstellen Sie eine dynamische Rechenknotengruppe mit der `purchaseOption` Einstellung auf `INTERRUPTIBLE_CAPACITY_RESERVATION` und der Startvorlage, die I-ODCR auf verweist. Setzen Sie die Mindestanzahl der Instanzen auf 0 und die maximale Anzahl auf 16 (entsprechend der I-ODCR Kapazität).

Überlegungen zur Abrechnung

- Der ODCR-Besitzer zahlt On-Demand Gebühren für ungenutzte Kapazität in den I-ODCR (Instances, die nicht vom Kunden gestartet wurden).
- Der Verbraucher zahlt On-Demand Tarife nur für Instances, die er tatsächlich startet und nutzt.

Weitere Informationen finden Sie unter Kapazitätsreservierung, Preise und Abrechnung im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit AWS 2 STÜCK

Amazon EC2 Capacity Blocks for ML ist eine Amazon EC2-Kaufoption, die es Ihnen ermöglicht, im Voraus zu zahlen, um GPU-based beschleunigte Recheninstanzen innerhalb eines bestimmten Datums- und Zeitbereichs für die Unterstützung kurzzeitiger Workloads zu reservieren. Instances, die innerhalb eines Capacity Blocks ausgeführt werden, werden innerhalb von Amazon EC2 automatisch nahe beieinander platziert, um Netzwerke mit niedriger Latenz UltraClusters, Petabit-Größe und ohne Blockierung zu ermöglichen. Weitere Informationen finden Sie unter Capacity Blocks for ML im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Sie können eine Startvorlage verwenden, damit AWS PCS beim Starten von Instances für eine Compute-Knotengruppe einen Capacity Block verwendet.

> **Note**
>
> AWS PCS hat seit Slurm-Version 24.05 die Unterstützung für Capacity Blocks eingeführt.

Einschränkungen

- Sie können eine Rechenknotengruppe nur einem Kapazitätsblock gleichzeitig zuordnen.
- Sie können eine Rechenknotengruppe nicht einer Kapazitätsreservierungsgruppe zuordnen, die mehrere Kapazitätsblöcke kombiniert.
- Kapazitätsblöcke müssen den `active` Status `scheduled` oder haben, um mit AWS PCS verwendet werden zu können. In anderen Bundesstaaten können Sie Kapazitätsblöcke nicht verwenden, z. `payment-failed` B. Weitere Informationen finden Sie unter „Kapazitätsblöcke anzeigen“ im Amazon Elastic Compute Cloud-Benutzerhandbuch.
- Informationen zu den Instance-Typen P6 und P5 finden Sie in der entsprechenden AWS-Dokumentation: Softwareanforderungen für P6-Instances, Maximieren Sie die Netzwerkbandbreite auf Amazon EC2-Instances mit mehreren Netzwerkkarten

Ablauf des Kapazitätsblocks

Kapazitätsblöcke sind auf einen bestimmten Datums- und Zeitbereich begrenzt. Wenn ein Kapazitätsblock abläuft:

- Die diesem Kapazitätsblock zugeordnete Rechenknotengruppe existiert weiterhin und bleibt denselben Warteschlangen zugeordnet.
- Alle Instances in der Compute-Knotengruppe werden beendet und aktive Jobs könnten je nach Ihren Slurm-Einstellungen fehlschlagen.
- AWS PCS kann keine neuen Instances in der Compute-Knotengruppe starten.
- Alle in der Warteschlange befindlichen oder neu übermittelten Jobs bleiben im Status Ausstehend, bis eine weitere Rechenknotengruppe an die Warteschlange angehängt wird oder Sie die Rechenknotengruppe so aktualisieren, dass sie eine neue Startvorlage verwendet, die einen neuen Kapazitätsblock angibt.

Konfigurieren Sie eine AWS PCS-Rechenknotengruppe für die Verwendung eines Kapazitätsblocks

Um einen Kapazitätsblock einer Rechenknotengruppe zuzuordnen

1. Erstellen Sie eine Amazon EC2 EC2-Startvorlage für AWS PCS, die Ihren Kapazitätsblock spezifiziert. Weitere Informationen zum Erstellen einer Startvorlage für AWS PCS finden Sie unterVerwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS.

   Ihre Startvorlage muss Folgendes enthalten:

   - Der Wert `MarketType` von `InstanceMarketOptions` muss auf gesetzt sein`capacity-block`.
   - A `CapacityReservationSpecification` mit einem gültigen `CapacityReservationId`
   - Ein gültiger `InstanceType` Wert, der dem Instance-Typ des von Ihnen gekauften Kapazitätsblocks entspricht.
2. Erstellen Sie eine Compute-Knotengruppe, die die Startvorlage verwendet. Weitere Informationen finden Sie unter Erstellen einer Compute-Knotengruppe in AWS STCK. Sie können auch eine bestehende Compute-Knotengruppe aktualisieren, um die Startvorlage zu verwenden. Weitere Informationen finden Sie unter Aktualisierung einer AWS PCS-Rechenknotengruppe.

   Wenn Sie die Compute-Knotengruppe erstellen oder aktualisieren:

   - Die IAM-Identität, die Sie zum Erstellen oder Aktualisieren der Compute-Knotengruppe verwenden, muss über die folgenden Berechtigungen verfügen:

     ```
     ec2:DescribeCapacityReservations
     ec2:DescribeCapacityBlocks
     ec2:DescribeCapacityBlockStatus
     ```

     Weitere Informationen finden Sie unter Mindestberechtigungen für AWS 5 STÜCK.
   - Der Kapazitätsblock muss sich im `active` Status `scheduled` oder befinden.
   - Stellen Sie den `purchaseOption` Wert der Compute-Knotengruppe auf ein`CAPACITY_BLOCK`.
   - Der Wert `maxInstanceCount` der Rechenknotengruppe darf die Größe des Kapazitätsblocks nicht überschreiten.

- Die Verfügbarkeitszone der Compute-Knotengruppe muss mit einer der Subnetz-Verfügbarkeitszonen der Compute-Knotengruppe übereinstimmen.

> **Important**
>
> Sie können den Instanztyp einer Compute-Knotengruppe nicht ändern, wenn Sie sie aktualisieren. Sie können einen Kapazitätsblock nur mit demselben Instanztyp wie die Compute-Knotengruppe verwenden. Wenn Sie einen Kapazitätsblock mit einem anderen Instanztyp verwenden möchten, müssen Sie eine neue Rechenknotengruppe erstellen.

UltraServer Instanzen konfigurieren

UltraServer NVIDIA-Instanzen verbinden mithilfe von NVLink mehrere GPUs über Instanzen hinweg. GPU-to-GPU Für die Kommunikation über NVLink muss der NVIDIA IMEX-Dienst mit den IP-Adressen aller Rechenknoten im Job konfiguriert werden. Auf dieser Seite erfahren Sie, wie Sie ein Slurm-Prolog-Skript erstellen, das IMEX automatisch für jeden Job konfiguriert, und wie Sie es in Ihrem PCS-Cluster bereitstellen. AWS

Voraussetzungen

- Ein AWS PCS-Cluster mit Slurm 25.05 oder höher.
- Eine Rechenknotengruppe, die UltraServer Instanzen mit einem Kapazitätsblock verwendet. Weitere Informationen finden Sie unter Konfigurieren Sie eine AWS PCS-Rechenknotengruppe für die Verwendung eines Kapazitätsblocks.
- Ein AMI mit den installierten NVIDIA-Treiber-, CUDA- und IMEX-Paketen. Der IMEX-Dienst (`nvidia-imex`) muss auf dem AMI aktiviert sein.
- Das Prolog-Skript muss auf allen Rechenknoten unter dem Pfad verfügbar sein, der in der `Prolog` Slurm-Einstellung des Clusters konfiguriert ist. Sie können ein gemeinsam verwendetes Dateisystem verwenden oder das Skript in Ihr benutzerdefiniertes AMI aufnehmen.

Automatische Topologieauswahl

Wenn Sie einen AWS PCS-Cluster mit UltraServer Rechenknotengruppen erstellen, überprüft das System den Kapazitätsblock, identifiziert den UltraServer Typ und konfiguriert Slurm mit dem entsprechenden Topologie-Plugin. Dieser Prozess läuft automatisch und erfordert keine Konfiguration.

AWS PCS verwaltet die Topologie über eine dynamisch generierte `topology.yaml` Datei. Während sich der Cluster durch Hinzufügen oder Entfernen von Rechenknotengruppen weiterentwickelt, gleicht AWS PCS die Topologiekonfiguration kontinuierlich ab, um den aktuellen Cluster-Status widerzuspiegeln.

Die Blocktopologie modelliert einheitliche Kommunikationsdomänen mit hoher Bandbreite, in denen alle GPUs an einer einzigen Hochgeschwindigkeits-NVLink-Domäne mit nahezu einheitlicher Latenz beteiligt sind. Das `topology/block` Plugin weist Knoten auf der Grundlage der definierten Netzwerktopologie zu und stellt so sicher, dass die Jobs innerhalb derselben Topologie platziert werden UltraServer und die GPU-to-GPU Kommunikation NVLink verwendet, anstatt auf das EFA-Netzwerk zurückzugreifen.

topology/block Das Plugin verwenden

Jobs, die an UltraServer Warteschlangen gesendet werden, werden vom Slurm-Scheduler automatisch innerhalb desselben Blocks (UltraServer) platziert. Alle Knoten in einem Block werden einem Job zugewiesen, bevor der nächste Block verwendet wird. Jobs müssen das `--exclusive` Flag verwenden, da IMEX pro Auftrag eine dedizierte IMEX-Domain benötigt und ein Neustart des IMEX-Dienstes alle anderen laufenden Jobs auf denselben Knoten stören würde. Weitere Informationen zum topology/block Plugin finden Sie im Topology-Handbuch und in der Referenz topology.yaml in der Slurm-Dokumentation.

Beim Senden von Aufträgen können Sie das `--segment=N` Argument with `sbatch` und die `srun` Befehle verwenden, um die Anzahl der Knoten anzugeben, die innerhalb desselben Blocks gruppiert werden sollen. Die Größe des Segments muss kleiner oder gleich der Größe des Planungsblocks sein. Die Verwendung garantiert `--segment` nicht, dass Segmente auf verschiedenen Blöcken platziert werden.

Im Folgenden finden Sie Beispielszenarien für die Zuweisung von Blöcken in einer Warteschlange mit 18 Knoten UltraServers (z. B. eine `u-p6e-gb200x72` mit 18 `p6e-gb200.36xlarge` Instanzen):

```
# Allocate a whole block (single UltraServer)
sbatch -N18 --exclusive --partition=ultra job.sh

# Allocate two full blocks (two UltraServers)
sbatch -N36 --exclusive --partition=ultra job.sh

# Allocate 24 nodes: full block is filled before 2nd block is used
sbatch -N24 --exclusive --partition=ultra job.sh

# Allocate 24 nodes as segments of 12, segments can fit on blocks with available space
```

```
sbatch -N24 --segment=12 --exclusive --partition=ultra job.sh
```

Bewährte Methoden für die Topologie UltraServer

Für eine optimale Leistung mit einer UltraServer Architektur in AWS PCS:

- Verwenden Sie Segmente für eine bessere Verfügbarkeit: Verwenden Sie die `--segment` Option mit einem Wert, der kleiner als die volle Blockgröße ist, damit Jobs auf Blöcken geplant werden können, die einige überlastete oder nicht verfügbare Knoten haben. Dabei wird die absolute NVLink-Lokalität gegen eine schnellere Auftragsplanung eingetauscht. Weitere Informationen finden Sie unter Mehr Kontrolle über die Knotenplanung mit dem Topology/Block Plugin.
- Berücksichtigen Sie die Auftragsgröße und die Blockgröße: Wenn`BlockSizes=18`, werden Jobs mit bis zu 18 Instanzen immer auf einer einzigen Instanz ausgeführt UltraServer. Jobs, die mehr als 18 Knoten erfordern, erstrecken sich über mehrere UltraServers, wobei NVLink-Konnektivität innerhalb jedes Blocks und EFA-Netzwerke zwischen den Blöcken vorhanden sind.
- Kompromisse bei Segmenten: Bei Verwendung der Option wird die Planung höherer Blockgrößen `--segment` deaktiviert. Ohne diese Option wird ein ganzer Block gefüllt`--segment`, bevor der nächste Block verwendet wird. Mit `--segment` können Segmente auf jedem Block mit ausreichend verfügbaren Knoten platziert werden.

Einschränkungen

- AWS PCS erfordert, dass UltraServer Rechenknotengruppen in dedizierten Warteschlangen platziert werden, getrennt von Gruppen, die keine UltraServer Rechenknoten sind.
- Jede Rechenknotengruppe kann nur einem UltraServer Kapazitätsblock zugeordnet werden.
- Wenn eine Warteschlange mehrere UltraServer Rechenknotengruppen enthält, muss die Instanzanzahl jeder Rechenknotengruppe ein Verhältnis von Zweierpotenzen aufweisen (z. B. 9 und 18 oder 2 und 8).
- Jobs in UltraServer Warteschlangen müssen das `--exclusive` Flag verwenden, um sicherzustellen, dass pro Auftrag eine dedizierte IMEX-Domain vorhanden ist.
- Die vom Prolog-Skript erstellte IMEX-Domäne umfasst nur die Knoten, die dem Job zugewiesen sind. Da AWS PCS den Aufträgen dynamisch Knoten zuweist, kann die IMEX-Domäne nicht vorab erstellt werden und muss zum Zeitpunkt des Auftragsstarts konfiguriert werden. Jobs, die sich über mehrere Bereiche UltraServers erstrecken, verfügen nicht über NVLink-Konnektivität zwischen Knoten in verschiedenen Knoten, UltraServers da die IMEX-Domäne auf IP-Adressen basiert und keine Brücke über separate NVLink-Fabrics herstellen kann.

- Die IMEX `nodes_config.cfg` kann nicht geändert werden, während der Dienst ausgeführt wird. Wenn die Konfiguration geändert wird, blockiert IMEX dauerhaft die Kommunikation zu Knoten, die nicht übereinstimmen. Die einzige Wiederherstellung besteht darin, IMEX auf allen Knoten zu stoppen und mit einer konsistenten Konfiguration neu zu starten.

Erstellen Sie das IMEX-Prolog-Skript

Das folgende `91_nvidia_imex_prolog.sh` Skript konfiguriert NVIDIA IMEX auf Rechenknoten, wenn ein Slurm-Job gestartet wird. Der Skriptname verwendet ein 91 Präfix, das der Benennungskonvention von SchedMD für die Reihenfolge der Prolog-Ausführung entspricht.

Das Skript führt die folgenden Aktionen aus:

1. Prüft, ob der Job exklusiv ist. Die IMEX-Konfiguration erfordert exklusive Jobs, um zu verhindern, dass andere laufende Jobs auf denselben Knoten unterbrochen werden.
2. Prüft, ob der Instanztyp ein unterstützter UltraServer Typ ist (`p6e-gb200`oder`p6e-gb300`).
3. Prüft, ob der IMEX-Dienst aktiviert ist.
4. Erstellt das IMEX-Standardkanalgerät, falls es nicht existiert.
5. Schreibt die privaten IP-Adressen aller Job-Nodes in`/etc/nvidia-imex/nodes_config.cfg`.
6. Startet den IMEX-Dienst neu.

> **Important**
>
> Für dieses Prolog-Skript müssen Jobs mit dem Flag eingereicht werden. `--exclusive` Wenn Sie IMEX neu starten, während andere Jobs auf denselben Knoten ausgeführt werden, werden diese Jobs unterbrochen.

> **Note**
>
> Dieses Prolog-Skript unterstützt `p6e-gb200` und Instanztypen. `p6e-gb300` Um es an andere Instanztypen anzupassen, aktualisieren Sie das `ALLOWED_INSTANCE_TYPES` Muster im Skript.

Speichern Sie das folgende Skript unter`91_nvidia_imex_prolog.sh`:

```
#!/usr/bin/env bash

# This prolog script configures NVIDIA IMEX on compute nodes involved in the job
 execution.
#
# In particular:
# - Checks whether the job is executed exclusively.
#   If not, it exits immediately because it requires jobs to be executed exclusively.
# - Checks if it is running on a supported UltraServer instance type.
#   If not, it exits immediately because IMEX must be configured only on those instance
 types.
# - Checks if the IMEX service is enabled.
#   If not, it exits immediately because IMEX must be enabled to get configured.
# - Creates the IMEX default channel.
#   For more information about IMEX channels, see
#   https://docs.nvidia.com/multi-node-nvlink-systems/imex-guide/imexchannels.html
# - Writes the private IP addresses of compute nodes into /etc/nvidia-imex/
nodes_config.cfg.
# - Restarts the IMEX system service.
#
# REQUIREMENTS:
#  - This prolog assumes to be run only with exclusive jobs.

LOG_FILE_PATH="/var/log/nvidia-imex-prolog.log"
SCONTROL_CMD="scontrol"
IMEX_START_TIMEOUT=60
IMEX_STOP_TIMEOUT=15
ALLOWED_INSTANCE_TYPES="^(p6e-gb200|p6e-gb300)"
IMEX_SERVICE="nvidia-imex"
IMEX_NODES_CONFIG="/etc/nvidia-imex/nodes_config.cfg"

function info() {
  echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [INFO] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function warn() {
  echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [WARN] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}

function error() {
  echo "$(date "+%Y-%m-%dT%H:%M:%S.%3N") [ERROR] [PID:$$] [JOB:${SLURM_JOB_ID}] $1"
}
```

```
function prolog_end() {
    info "PROLOG End JobId=${SLURM_JOB_ID}: $0"
    info "----------------"
    exit 0
}

function get_instance_type() {
  local token=$(curl -X PUT -s "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600")
  curl -s -H "X-aws-ec2-metadata-token: ${token}" http://169.254.169.254/latest/meta-
data/instance-type
}

function return_if_unsupported_instance_type() {
  local instance_type=$(get_instance_type)

  if [[ ! ${instance_type} =~ ${ALLOWED_INSTANCE_TYPES} ]]; then
    info "Skipping IMEX configuration because instance type ${instance_type} does not
 support it"
    prolog_end
  fi
}

function return_if_imex_disabled() {
  if ! systemctl is-enabled "${IMEX_SERVICE}" &>/dev/null; then
    warn "Skipping IMEX configuration because system service ${IMEX_SERVICE} is not
 enabled"
    prolog_end
  fi
}

function return_if_job_is_not_exclusive() {
  if [[ "${SLURM_JOB_OVERSUBSCRIBE}" =~ ^(NO|TOPO)$  ]]; then
    info "Job is exclusive, proceeding with IMEX configuration"
  else
    info "Skipping IMEX configuration because the job is not exclusive"
    prolog_end
  fi
}

function get_ips_from_node_names() {
  local _nodes=$1
  ${SCONTROL_CMD} -ao show node "${_nodes}" | sed 's/^.* NodeAddr=\([^ ]*\).*/\1/'
}
```

```
function reload_imex() {
  info "Stopping IMEX"
  timeout ${IMEX_STOP_TIMEOUT} systemctl stop ${IMEX_SERVICE}
  pkill -9 ${IMEX_SERVICE}

  info "Restarting IMEX"
  timeout ${IMEX_START_TIMEOUT} systemctl start ${IMEX_SERVICE}
}

function create_default_imex_channel() {
  info "Creating IMEX default channel"
  MAJOR_NUMBER=$(cat /proc/devices | grep nvidia-caps-imex-channels | cut -d' ' -f1)
  if [ ! -d "/dev/nvidia-caps-imex-channels" ]; then
    sudo mkdir /dev/nvidia-caps-imex-channels
  fi

  # Then check and create device node
  if [ ! -e "/dev/nvidia-caps-imex-channels/channel0" ]; then
    sudo mknod /dev/nvidia-caps-imex-channels/channel0 c $MAJOR_NUMBER 0
    info "IMEX default channel created"
  else
    info "IMEX default channel already exists"
  fi
}

{
  info "PROLOG Start JobId=${SLURM_JOB_ID}: $0"

  return_if_job_is_not_exclusive
  return_if_unsupported_instance_type
  return_if_imex_disabled

  create_default_imex_channel

  IPS_FROM_CR=$(get_ips_from_node_names "${SLURM_NODELIST}")

  info "Node Names: ${SLURM_NODELIST}"
  info "Node IPs: ${IPS_FROM_CR}"
  info "IMEX Nodes Config: ${IMEX_NODES_CONFIG}"

  info "Updating IMEX nodes config ${IMEX_NODES_CONFIG}"
  echo "${IPS_FROM_CR}" > "${IMEX_NODES_CONFIG}"
  reload_imex
```

```
  prolog_end

} 2>&1 | tee -a "${LOG_FILE_PATH}" | logger -t "91_nvidia_imex_prolog"
```

Stellen Sie das Prolog-Skript bereit

Um das Prolog-Skript bereitzustellen, konfigurieren Sie den Slurm `Prolog` und die `PrologFlags` benutzerdefinierten Einstellungen in Ihrem Cluster. Sie können diese festlegen, wenn Sie einen Cluster erstellen oder einen vorhandenen Cluster aktualisieren.

Weitere Informationen zur Konfiguration benutzerdefinierter Slurm-Einstellungen finden Sie unterBenutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster. Das Verfahren zum Aktualisieren eines Clusters finden Sie unterAktualisiere ein AWS PCS-Cluster.

Legen Sie die folgenden benutzerdefinierten Slurm-Einstellungen auf dem Cluster fest:

- `Prolog`— Stellen Sie den Pfad ein, in dem das Prolog-Skript beispielsweise auf Rechenknoten verfügbar ist. `/shared/scripts/prolog.d/91_nvidia_imex_prolog.sh`
- `PrologFlags`- Auf setzen. `Alloc,NoHold` Das `Alloc` Flag führt das Prolog auf jedem Knoten aus, wenn dieser zum ersten Mal einem Job zugewiesen wird. Das `NoHold` Flag verhindert, dass der Job angehalten wird, wenn der Prolog fehlschlägt.

Example Aktualisieren Sie den Cluster mit den Prolog-Einstellungen mithilfe der AWS CLI

```
aws pcs update-cluster \
  --cluster-identifier <cluster-id> \
  --slurm-configuration '{
    "slurmCustomSettings": [
      {"parameterName": "Prolog", "parameterValue": "/shared/scripts/
prolog.d/91_nvidia_imex_prolog.sh"},
      {"parameterName": "PrologFlags", "parameterValue": "Alloc,NoHold"}
    ]
  }'
```

Validieren Sie das IMEX-Setup

Das Prolog-Skript wird automatisch ausgeführt, wenn Sie einen Slurm-Job einreichen. Senden Sie einen Testjob, um zu überprüfen, ob IMEX auf allen Knoten korrekt konfiguriert ist.

Erstellen Sie ein Job-Skript mit dem Namen`check-imex.sh`:

```
#!/bin/bash
#SBATCH --job-name=nvidia-imex-status
#SBATCH --ntasks-per-node=1
#SBATCH --exclusive
#SBATCH --output=imex-status-%j.out
#SBATCH --error=imex-status-%j.err

srun bash -c "/usr/bin/nvidia-imex-ctl -N > imex_result_\${SLURM_JOB_ID}_\
$(hostname).out 2>&1"
```

Reichen Sie den Job in Ihre UltraServer Warteschlange ein:

```
sbatch -p queue-name -N node-count check-imex.sh
```

Überprüfen Sie die Ausgabedateien. Eine erfolgreiche IMEX-Konfiguration zeigt in der Konnektivitätstabelle alle Knoten im `READY` Status `C` (Verbunden) an und`Domain State: UP`:

```
Connectivity Table Legend:
I - Invalid - Node wasn't reachable, no connection status available
N - Never Connected
R - Recovering - Connection was lost, but clean up has not yet been triggered.
D - Disconnected - Connection was lost, and clean up has been triggreed.
A - Authenticating - If GSSAPI enabled, client has initiated mutual authentication.
!V! - Version mismatch, communication disabled.
!M! - Node map mismatch, communication disabled.
C - Connected - Ready for operation

5/4/2026 06:08:10.580
Nodes:
Node #0   - 172.31.48.81    - READY               - Version: 570.172
Node #1   - 172.31.48.98    - READY               - Version: 570.172
Node #2   - 172.31.48.221   - READY               - Version: 570.172
Node #3   - 172.31.49.228   - READY               - Version: 570.172
Node #4   - 172.31.50.39    - READY               - Version: 570.172
Node #5   - 172.31.50.44    - READY               - Version: 570.172
Node #6   - 172.31.51.66    - READY               - Version: 570.172
Node #7   - 172.31.51.157   - READY               - Version: 570.172
Node #8   - 172.31.52.239   - READY               - Version: 570.172
Node #9   - 172.31.53.80    - READY               - Version: 570.172
Node #10  - 172.31.54.95    - READY               - Version: 570.172
Node #11  - 172.31.54.183   - READY               - Version: 570.172
Node #12  - 172.31.54.203   - READY               - Version: 570.172
```

```
Node #13  - 172.31.54.241   - READY                - Version: 570.172
Node #14  - 172.31.55.59    - READY                - Version: 570.172
Node #15  - 172.31.55.187   - READY                - Version: 570.172
Node #16  - 172.31.55.197   - READY                - Version: 570.172
Node #17  - 172.31.56.47    - READY                - Version: 570.172

 Nodes From\To  0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17
       0        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       1        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       2        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       3        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       4        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       5        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       6        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       7        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       8        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
       9        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      10        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      11        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      12        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      13        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      14        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      15        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      16        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C
      17        C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C   C

Domain State: UP
```

Fehlerbehebung

Die Zuordnung der Knotenkonfiguration stimmt nicht überein

Wenn Sie `!M!` in der Konnektivitätstabelle oder `Node configuration mapping mismatch` in den IMEX-Protokollen feststellen, ist die `nodes_config.cfg` Datei nicht auf allen Knoten konsistent. IMEX blockiert dauerhaft die Kommunikation mit Knoten, die nicht übereinstimmen. Stellen Sie sicher, dass das Prolog-Skript auf allen Knoten ausgeführt wurde, und starten Sie den Job neu.

Das IMEX-Kanalgerät wurde nicht gefunden

Wenn das Prolog-Skript das IMEX-Kanalgerät nicht erstellen kann, überprüfen Sie, ob der NVIDIA-Treiber installiert ist und Folgendes enthält. `/proc/devices` `nvidia-caps-imex-channels`

Das Prolog-Skript wird nicht ausgeführt

Stellen Sie sicher, dass die `PrologFlags` Einstellungen `Prolog` und auf dem Cluster konfiguriert sind. Stellen Sie sicher, dass die Skriptdatei auf den Rechenknoten vorhanden ist und über Ausführungsberechtigungen verfügt (`chmod 0755`).

Häufig gestellte Fragen zur Verwendung von Capacity Blocks mit AWS PCS

Ich habe gerade für einen Kapazitätsblock bezahlt und sofort versucht, ihn mit AWS PCS zu verwenden, aber die Erstellung der Compute-Knotengruppe ist fehlgeschlagen. Was ist passiert?

Ihr Kapazitätsblock befindet sich möglicherweise nicht im `active` Status `scheduled` Oder. Versuchen Sie es erneut, wenn der Kapazitätsblock den Wert `scheduled` oder hat`active`.

Ich verwende einen Capacity Block in AWS PCS und habe eine Erweiterung gekauft, bevor sie abgelaufen ist. Wie verwende ich ihn weiterhin in AWS PCS?

Sie müssen nichts tun, um den Capacity Block in AWS PCS weiterhin zu verwenden. Das Enddatum Ihres Capacity Blocks wird aktualisiert, sobald Ihre Verlängerungszahlung erfolgreich war. Solange Ihr Kapazitätsblock nicht abläuft, ist die Rechenknotengruppe weiterhin in Betrieb. Wenn Ihre Verlängerungszahlung fehlschlägt, bleibt Ihr Kapazitätsblock bestehen `active` und die Rechenknotengruppe funktioniert, bis der Kapazitätsblock an seinem ursprünglichen Enddatum abläuft.

Was passiert mit meinen in der Warteschlange stehenden und laufenden Jobs, wenn mein Kapazitätsblock abläuft?

Jobs in der Warteschlange, die nicht gestartet wurden, bevor der Kapazitätsblock abgelaufen ist, bleiben solange ausstehend, bis Sie eine weitere Rechenknotengruppe an die Warteschlange anhängen oder die Rechenknotengruppe mit einem neuen Kapazitätsblock aktualisieren. Sie können weiterhin Jobs an die Warteschlange senden. Ihre Slurm-Einstellungen wirken sich auf aktive Jobs aus. Standardmäßig werden aktive Jobs automatisch erneut in die Warteschlange gestellt, können aber Fehler aufweisen oder fehlschlagen.

Mein Kapazitätsblock ist abgelaufen. Sollte ich etwas tun?

Du musst nichts tun. Sie können in der Amazon EC2 EC2-Konsole den Status Ihrer EC2-Kapazitätsreservierungen überprüfen. Wenn ein Kapazitätsblock abläuft, ist die diesem Kapazitätsblock zugeordnete Rechenknotengruppe weiterhin vorhanden und verarbeitet dieselben Warteschlangen. Die Rechenknotengruppe hat keine Instanzen zum Ausführen von Jobs. Sie

können die Compute-Knotengruppe löschen oder sie von den Warteschlangen trennen, um zu verhindern, dass Benutzer Jobs einreichen, die nicht ausgeführt werden können.

Ich möchte einen neuen Kapazitätsblock mit meiner AWS PCS-Compute-Knotengruppe verwenden. Was soll ich tun?

Wir empfehlen Ihnen, eine neue Rechenknotengruppe zu erstellen, um den neuen Kapazitätsblock zu verwenden. Weitere Informationen finden Sie unter Konfigurieren Sie eine AWS PCS-Rechenknotengruppe für die Verwendung eines Kapazitätsblocks.

Wie kann ich einen Kapazitätsblock für Cluster und Dienste gemeinsam nutzen?

Sie können einen Kapazitätsblock auf mehrere Cluster und Dienste aufteilen. Um beispielsweise einen Kapazitätsblock mit 64 `p5.48xlarge` Instanzen mit 20 Knoten auf PCS-Cluster-1, 16 Knoten auf PCS-Cluster-2 und den verbleibenden Knoten für andere Dienste aufzuteilen, setzen Sie beide und auf 20 für PCS-Cluster-1 `minInstanceCount` und `maxInstanceCount` 16 für PCS-Cluster-2.

Kann ich mehr als einen Kapazitätsblock oder kombinierte Kapazität mit einer Rechenknotengruppe verwenden?

Nein. Einer einzelnen Rechenknotengruppe kann nur 1 Kapazitätsblock zugeordnet werden. AWS PCS unterstützt keine Kapazitätsreservierungsgruppen, die mehrere Kapazitätsblöcke kombinieren.

Woher weiß ich, wann meine Kapazitätsblöcke beginnen oder ablaufen?

Unabhängig von AWS PCS sendet Amazon EC2 ein `Capacity Block Reservation Delivered` Ereignis, EventBridge wenn eine Kapazitätsblock-Reservierung beginnt, und ein `Capacity Block Reservation Expiration Warning` Ereignis 40 Minuten vor Ablauf der Kapazitätsblock-Reservierung. Weitere Informationen finden Sie unter Verwendung von Kapazitätsblöcken überwachen EventBridge im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Wie verfolgt Slurm den Status meines Kapazitätsblocks?

Du kannst laufen`sinfo`, um zu verstehen, wie AWS PCS den Capacity Block verwendet. In der folgenden Beispielausgabe ist eine Warteschlange einer Rechenknotengruppe zugeordnet, die 4 Instanzen aus einem `active` Kapazitätsblock ausführt. Die Knoten befinden sich im `idle` Slurm-Status (zur Verwendung verfügbar und noch keinen Jobs zugewiesen).

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
```

```
fanout up infinite 4 idle node-fanout-[1-4]
```

Wenn sich die Knoten stattdessen im `maint` Status befinden, können Sie den Befehl ausführen, `scontrol show res` um Details zur Slurm-Reservierung zu sehen, die diesen Status kontrolliert. In der folgenden Beispielausgabe hat der `scheduled` Capacity-Block ein future Startdatum.

```
$ scontrol show res

ReservationName=node-fanout-scheduled StartTime=2025-10-14T13:09:17
 EndTime=2025-10-14T13:11:17 Duration=00:02:00
   Nodes=node-fanout-[1-4] NodeCnt=4 CoreCnt=16 Features=(null) PartitionName=(null)
 Flags=MAINT,SPEC_NODES
   TRES=cpu=16

   Users=root Groups=(null) Accounts=(null) Licenses=(null) State=ACTIVE
 BurstBuffer=(null)
   MaxStartDelay=(null)

   Comment=node-fanout Scheduled
```

Wie kann ich feststellen, ob die Fehler, die ich beim Starten von Capacity erhalte, darauf zurückzuführen sind, dass mein Capacity-Block gemeinsam genutzt wird?

Überprüfen Sie Capacity Reservations in der Amazon EC2 EC2-Konsole, um herauszufinden, wie viele Instances aus dem Capacity Block aktiv bereitgestellt werden. Überprüfen Sie die Tags der einzelnen Instances, um herauszufinden, welcher Service oder Cluster sie verwendet. Beispielsweise verfügen alle Instanzen für AWS PCS über AWS PCS-Tags, `aws:pcs:cluster-id = pcs_l0mizqyk5o | aws:pcs:compute-node-group-id = pcs_ic7onkmfqk` die angeben, zu welchen Clustern und Rechenknotengruppen die Instanz gehört. Sie können dann überprüfen, ob der Kapazitätsblock die maximale Kapazität erreicht hat.

Sie verwenden`scontrol show nodes`, um zu überprüfen, ob ein Capacity Block-Knoten in einem AWS PCS-Cluster Folgendes auslöst`ReservationCapacityExceeded`:

```
[root@ip-172-16-10-54 ~]# scontrol show nodes test-node-8-gamma-cb-2
NodeName=test-8-gamma-cb-2 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=8 CPUTot=8 CPULoad=0.00
   AvailableFeatures=test-8-gamma-cb,gpu
   ActiveFeatures=test-8-gamma-cb,gpu
   Gres=gpu:H100:1
```

```
   NodeAddr=test-8-gamma-cb-2 NodeHostName=test-8-gamma-cb-2
   RealMemory=249036 AllocMem=0 FreeMem=N/A Sockets=8 Boards=1
   State=IDLE+CLOUD+POWERING_DOWN ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A
 MCS_label=N/A
   Partitions=my-q
   BootTime=None SlurmdStartTime=None
   LastBusyTime=Unknown ResumeAfterTime=None
   CfgTRES=cpu=8,mem=249036M,billing=8
   AllocTRES=
   CurrentWatts=0 AveWatts=0
   Reason=Failed to launch backing instance (Error Code:
 ReservationCapacityExceeded) [root@2025-08-28T15:15:33]
```

Wie kann ich erzwingen, dass ein Job auf Capacity Block-gestützten Instances ausgeführt wird, wenn mehrere Rechenknotengruppen an dieselbe Warteschlange angehängt sind?

Sie können die Funktionen und Einschränkungen von Slurm verwenden, um einen Job an eine bestimmte Gruppe von Knoten zu binden. Wir empfehlen, Slurm-Gewichtungen nicht für jede Rechenknotengruppe festzulegen, da dies nur mit Knoten funktioniert, die sich nicht im `maint` Status befinden.

Nützliche Parameter für Startvorlagen

In diesem Abschnitt werden einige Parameter für Startvorlagen beschrieben, die für AWS PCS allgemein nützlich sein können.

Aktivieren Sie die detaillierte CloudWatch Überwachung

Mithilfe eines Startvorlagenparameters können Sie die Erfassung von CloudWatch Metriken in kürzeren Intervallen aktivieren.

AWS-Managementkonsole

Auf den Konsolenseiten zum Erstellen oder Bearbeiten von Startvorlagen befindet sich diese Option im Abschnitt Erweiterte Details. Stellen Sie „Detaillierte CloudWatch Überwachung“ auf „Aktivieren“.

YAML

```
Monitoring:
    Enabled: True
```

JSON

```
{"Monitoring": {"Enabled": "True"}}
```

Weitere Informationen finden Sie unter Aktivieren oder Deaktivieren der detaillierten Überwachung für Ihre Instances im Amazon Elastic Compute Cloud-Benutzerhandbuch für Linux-Instances.

Instanz-Metadaten-Service Version 2 (IMDS v2)

Die Verwendung von IMDS v2 mit EC2-Instances bietet erhebliche Sicherheitsverbesserungen und trägt dazu bei, potenzielle Risiken im Zusammenhang mit dem Zugriff auf Instance-Metadaten in Umgebungen zu minimieren. AWS

AWS-Managementkonsole

Auf den Konsolenseiten zum Erstellen oder Bearbeiten von Startvorlagen befindet sich diese Option im Abschnitt Erweiterte Details. Stellen Sie für Metadaten, auf die zugegriffen werden kann, die Option Aktiviert, die Metadatenversion auf Nur V2 (Token erforderlich) und das Limit für den Metadaten-Response-Hop auf 4 ein.

YAML

```
MetadataOptions:
  HttpEndpoint: enabled
  HttpTokens: required
  HttpPutResponseHopLimit: 4
```

JSON

```
{
    "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpPutResponseHopLimit": 4,
        "HttpTokens": "required"
    }
}
```

.

AWS PCS-Warteschlangen

Eine AWS PCS-Warteschlange ist eine einfache Abstraktion gegenüber der systemeigenen Implementierung einer Arbeitswarteschlange durch den Scheduler. Im Fall von Slurm entspricht eine AWS PCS-Warteschlange einer Slurm-Partition.

Benutzer senden Jobs an eine Warteschlange, in der sie sich befinden, bis sie so geplant werden können, dass sie auf Knoten ausgeführt werden, die von einer oder mehreren Rechenknotengruppen bereitgestellt werden. Ein AWS PCS-Cluster 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.

Themen

- Eine Warteschlange in AWS PCS erstellen
- Aktualisierung einer AWS PCS-Warteschlange
- Löschen einer Warteschlange in AWS PCS

Eine Warteschlange in AWS PCS erstellen

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was beim Erstellen einer Warteschlange in AWS PCS zu beachten ist.

> **Note**
>
> Sie können benutzerdefinierte Slurm-Einstellungen für Warteschlangen konfigurieren, um partitionsspezifische Planungsrichtlinien und Ressourcenverwaltung zu implementieren. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK.

Voraussetzungen

- Ein AWS PCS-Cluster — Warteschlangen können nur in Verbindung mit einem bestimmten PCS-Cluster erstellt werden. AWS

- Eine oder mehrere AWS PCS-Compute-Knotengruppen — eine Warteschlange muss mindestens einer AWS PCS-Compute-Knotengruppe zugeordnet sein.

Um eine Warteschlange in AWS PCS zu erstellen

Sie können eine Warteschlange mit dem AWS-Managementkonsole oder dem erstellen AWS CLI.

AWS-Managementkonsole

Um eine Warteschlange mit der Konsole zu erstellen

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den Cluster für die Warteschlange aus. Navigieren Sie zu Warteschlangen und wählen Sie Warteschlange erstellen.
3. Geben Sie im Abschnitt Warteschlangenkonfiguration die folgenden Werte an:
    a. Warteschlangenname — Ein Name für Ihre Warteschlange. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht länger als 25 Zeichen sein. Der Name muss innerhalb des Clusters eindeutig sein.
    b. Compute-Knotengruppen — Wählen Sie eine oder mehrere Compute-Knotengruppen aus, um diese Warteschlange zu bedienen. Eine Rechenknotengruppe kann mehr als einer Warteschlange zugeordnet werden.
4. (Optional) Im Abschnitt Zusätzliche Scheduler-Einstellungen können Sie Parameternamen- und Wertepaare hinzufügen, um zusätzliche Slurm-Einstellungen zu konfigurieren. Eine vollständige Liste der unterstützten Parameter finden Sie unter. Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen
5. (Optional) Fügen Sie unter Tags beliebige Tags zu Ihrer AWS PCS-Warteschlange hinzu
6. Wählen Sie Create queue (Warteschlange erstellen) aus. Im Statusfeld wird Creating angezeigt, während AWS PCS die Warteschlange erstellt. Die Erstellung der Warteschlange kann mehrere Minuten dauern.

Als nächster Schritt wird empfohlen

- Reichen Sie einen Job in Ihre neue Warteschlange ein.

AWS CLI

Um eine Warteschlange zu erstellen mit AWS CLI

Verwenden Sie den folgenden Befehl, um Ihre Warteschlange zu erstellen. Nehmen Sie die folgenden Ersetzungen vor:

1. *region-code*Ersetzen Sie durch die AWS Region des Clusters. Beispiel, `us-east-1`.
2. *my-queue*Ersetzen Sie es durch den Namen für Ihre Warteschlange. Der Name darf nur alphanumerische Zeichen (wobei die Groß- und Kleinschreibung beachtet werden muss) und Bindestriche enthalten. Er muss mit einem alphabetischen Zeichen beginnen und darf nicht länger als 25 Zeichen sein. Der Name muss innerhalb des Clusters eindeutig sein.
3. *my-cluster*Ersetzen Sie ihn durch den Namen oder die ID Ihres Clusters.
4. *compute-node-group-id*Ersetzen Sie es durch die ID der Rechenknotengruppe, die die Warteschlange bedienen soll. Beispiel, `pcs_abcdef12345`.

> **Note**
>
> Wenn Sie eine Warteschlange erstellen, müssen Sie die ID der Compute-Knotengruppe und nicht deren Namen angeben.

```
aws pcs create-queue --region region-code \
    --queue-name my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=compute-node-group-id
```

Example— Erstellen einer Warteschlange mit benutzerdefinierten Slurm-Einstellungen

```
aws pcs create-queue --region region-code \
    --queue-name my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=compute-node-group-id \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen.

Das Erstellen der Warteschlange kann mehrere Minuten dauern. Sie können den Status Ihrer Warteschlange mit dem folgenden Befehl abfragen. Sie können keine Jobs an die Warteschlange senden, bis ihr Status erreicht ist`ACTIVE`.

```
aws pcs get-queue --region region-code \
    --cluster-identifier my-cluster \
    --queue-identifier my-queue
```

Als nächster Schritt wird empfohlen

- Reichen Sie einen Job in Ihre neue Warteschlange ein

Aktualisierung einer AWS PCS-Warteschlange

Dieses Thema bietet einen Überblick über die verfügbaren Optionen und beschreibt, was bei der Aktualisierung einer AWS PCS-Warteschlange zu beachten ist. Informationen zu den benutzerdefinierten Slurm-Einstellungen finden Sie unterBenutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen.

Überlegungen beim Aktualisieren einer AWS PCS-Warteschlange

Warteschlangenaktualisierungen wirken sich nicht auf laufende Jobs aus, aber der Cluster kann möglicherweise keine neuen Jobs annehmen, während die Warteschlange aktualisiert wird.

Um eine AWS PCS-Warteschlange zu aktualisieren

Sie können das AWS-Managementkonsole oder verwenden AWS CLI , um eine Warteschlange zu aktualisieren.

AWS-Managementkonsole

Um eine Warteschlange zu aktualisieren

1. Öffnen Sie die AWS PCS-Konsole unter `https://console.aws.amazon.com/pcs/home#/clusters`
2. Wählen Sie den Cluster aus, in dem Sie eine Warteschlange aktualisieren möchten.

3. Navigieren Sie zu Warteschlangen, gehen Sie zu der Warteschlange, die Sie aktualisieren möchten, und wählen Sie dann Bearbeiten aus.
4. Aktualisieren Sie im Abschnitt Warteschlangenkonfiguration einen der folgenden Werte:
   - Knotengruppen — Fügen Sie Compute-Knotengruppen hinzu oder entfernen Sie sie aus der Zuordnung zur Warteschlange.
   - Zusätzliche Scheduler-Einstellungen — Fügen Sie benutzerdefinierte Slurm-Einstellungen für die Warteschlange hinzu, ändern oder entfernen Sie sie. Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen.
   - Tags — Fügen Sie Tags für die Warteschlange hinzu oder entfernen Sie sie.
5. Wählen Sie Aktualisieren aus. Im Feld Status wird die Meldung Aktualisierung angezeigt, während die Änderungen übernommen werden.

> **Important**
>
> Aktualisierungen in der Warteschlange können mehrere Minuten dauern.

AWS CLI

Um eine Warteschlange zu aktualisieren

1. Aktualisieren Sie Ihre Warteschlange mit dem folgenden Befehl. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:
   a. *region-code*Ersetzen Sie es durch AWS-Region das, in dem Sie Ihren Cluster erstellen möchten.
   b. *my-queue*Ersetzen Sie es durch den Namen oder `computeNodeGroupId` für Ihre Warteschlange.
   c. *my-cluster*Ersetzen Sie durch den Namen oder `clusterId` Ihres Clusters.
   d. Um die Zuordnungen von Compute-Knotengruppen zu ändern, stellen Sie eine aktualisierte Liste für bereit`--compute-node-group-configurations`.
      - Um beispielsweise eine zweite Compute-Knotengruppe hinzuzufügen`computeNodeGroupExampleID2`:

```
--compute-node-group-configurations
 computeNodeGroupId=computeNodeGroupExampleID1,computeNodeGroupId=computeNodeGro
```

```
aws pcs update-queue --region region-code \
    --queue-identifier my-queue \
    --cluster-identifier my-cluster \
    --compute-node-group-configurations \
    computeNodeGroupId=computeNodeGroupExampleID1
```

Example— Aktualisierung einer Warteschlange mit benutzerdefinierten Slurm-Einstellungen

```
aws pcs update-queue --region region-code \
    --queue-identifier my-queue \
    --cluster-identifier my-cluster \
    --slurm-configuration \
    'slurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen.

2. Das Aktualisieren der Warteschlange kann mehrere Minuten dauern. Sie können den Status Ihrer Warteschlange mit dem folgenden Befehl abfragen. Sie können keine Jobs an die Warteschlange senden, bis ihr Status erreicht ist`ACTIVE`.

```
aws pcs get-queue --region region-code \
    --cluster-identifier my-cluster \
    --queue-identifier my-queue
```

Empfohlene nächste Schritte

- Reichen Sie einen Job in Ihre aktualisierte Warteschlange ein.

Löschen einer Warteschlange in AWS PCS

Dieses Thema bietet einen Überblick über das Löschen einer Warteschlange in AWS PCS.

Überlegungen beim Löschen einer Warteschlange

- Wenn in der Warteschlange Jobs ausgeführt werden, werden sie vom Scheduler beendet, wenn die Warteschlange gelöscht wird. Ausstehende Jobs in der Warteschlange werden storniert. Erwägen Sie, darauf zu warten, dass Jobs in der Warteschlange abgeschlossen sind, oder stop/cancel sie manuell mit den systemeigenen Befehlen des Schedulers (z. B. `scancel` für Slurm) zu bearbeiten.

Lösche die Warteschlange

Sie können das AWS-Managementkonsole oder verwenden AWS CLI , um eine Warteschlange zu löschen.

AWS-Managementkonsole

So löschen Sie eine Warteschlange

1. Öffnen Sie die AWS PCS-Konsole.
2. Wählen Sie den Cluster der Warteschlange aus.
3. Navigieren Sie zu Warteschlangen und wählen Sie die Warteschlange aus, die Sie löschen möchten.
4. Wählen Sie Löschen aus.
5. Das Feld Status wird angezeigt`Deleting`. Das kann mehrere Minuten dauern.

> **Note**
>
> Sie können Befehle verwenden, die Ihrem Scheduler eigen sind, um zu bestätigen, dass die Warteschlange gelöscht wurde. Verwenden Sie zum Beispiel `sinfo` or `squeue` für Slurm.

AWS CLI

So löschen Sie eine Warteschlange

- Verwenden Sie den folgenden Befehl, um eine Warteschlange mit diesen Ersetzungen zu löschen:

- Ersetzen Sie *region-code* durch den, in dem sich AWS-Region Ihr Cluster befindet.
- *my-queue*Ersetzen Sie durch den Namen oder die ID Ihrer Warteschlange.
- *my-cluster*Ersetzen Sie durch den Namen oder die ID Ihres Clusters.

```
aws pcs delete-queue --region region-code \
       --queue-identifier my-queue \
       --cluster-identifier my-cluster
```

Das Löschen der Warteschlange kann mehrere Minuten dauern.

> **Note**
>
> Sie können Befehle verwenden, die Ihrem Scheduler eigen sind, um zu bestätigen, dass die Warteschlange gelöscht wurde. Verwenden Sie zum Beispiel `sinfo` or `squeue` für Slurm.

AWS PCS-Anmeldeknoten

Ein AWS PCS-Cluster benötigt normalerweise mindestens einen Anmeldeknoten, um den interaktiven Zugriff und die Auftragsverwaltung zu unterstützen. Eine Möglichkeit, dies zu erreichen, besteht darin, eine statische AWS PCS-Rechenknotengruppe zu verwenden, die für die Funktion eines Anmeldeknotens konfiguriert ist. Sie können auch eine eigenständige EC2-Instance so konfigurieren, dass sie als Anmeldeknoten fungiert.

Themen

- Verwendung einer AWS PCS-Compute-Knotengruppe zur Bereitstellung von Anmeldeknoten
- Verwenden eigenständiger Instanzen als AWS PCS-Anmeldeknoten
- Verbinden eines eigenständigen Anmeldeknotens mit mehreren Clustern in AWS STCK

Verwendung einer AWS PCS-Compute-Knotengruppe zur Bereitstellung von Anmeldeknoten

Dieses Thema bietet einen Überblick über vorgeschlagene Konfigurationsoptionen und beschreibt, was zu beachten ist, wenn Sie eine AWS PCS-Rechenknotengruppe verwenden, um dauerhaften, interaktiven Zugriff auf Ihren Cluster bereitzustellen.

Erstellen einer AWS PCS-Rechenknotengruppe für Anmeldeknoten

Operativ unterscheidet sich dies nicht wesentlich von der Erstellung einer regulären Rechenknotengruppe. Es müssen jedoch einige wichtige Konfigurationsentscheidungen getroffen werden:

- Legen Sie eine statische Skalierungskonfiguration für mindestens eine EC2-Instance in der Compute-Knotengruppe fest.
- Wählen Sie die On-Demand-Kaufoption, um zu vermeiden, dass Ihre Instance (s) zurückgefordert werden.
- Wählen Sie einen aussagekräftigen Namen für die Compute-Knotengruppe, z. B. Login.
- Wenn Sie möchten, dass auf die Login-Knoten-Instanz (en) außerhalb Ihrer VPC zugegriffen werden kann, sollten Sie die Verwendung eines öffentlichen Subnetzes in Betracht ziehen.
- Wenn Sie den SSH-Zugriff zulassen möchten, muss die Startvorlage über eine Sicherheitsgruppe verfügen, die den SSH-Port den IP-Adressen Ihrer Wahl zugänglich macht.

- Das IAM-Instance-Profil sollte nur die AWS-Berechtigungen haben, die Ihre Endbenutzer haben sollen. Details dazu finden Sie unter IAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung.
- Erwägen Sie, AWS Systems Manager Session Manager die Verwaltung Ihrer Login-Instances zu gestatten.
- Erwägen Sie, den Zugriff auf die AWS-Anmeldeinformationen der Instanz auf Administratorbenutzer zu beschränken
- Wählen Sie kostengünstigere Instance-Typen als für reguläre Compute-Knotengruppen aus, da die Login-Knoten kontinuierlich laufen.
- Verwenden Sie dasselbe (oder ein abgeleitetes) AMI wie für Ihre anderen Compute-Knotengruppen, um sicherzustellen, dass auf allen Instances dieselbe Software installiert ist. Weitere Informationen zum Anpassen finden Sie AMIs unter Amazon Machine Images (AMIs) für AWS STK.
- Konfigurieren Sie dasselbe Netzwerkdateisystem (Amazon EFS, Amazon FSx for Lustre usw.), das auf Ihren Anmeldeknoten bereitgestellt wird wie auf Ihren Compute-Instances. Weitere Informationen finden Sie unter Verwenden von Netzwerkdateisystemen mit AWS PCS.

Greifen Sie auf Ihre Anmeldeknoten zu

Sobald Ihre neue Compute-Knotengruppe den Status ACTIVE erreicht hat, können Sie die EC2-Instance (en) finden, die sie erstellt hat, und sich bei ihnen anmelden. Weitere Informationen finden Sie unter Suchen nach Rechenknotengruppeninstanzen in AWS PCS.

Aktualisierung einer AWS PCS-Compute-Knotengruppe für Login-Knoten

Sie können eine Anmeldeknotengruppe aktualisieren mit UpdateComputeNodeGroup. Im Rahmen des Aktualisierungsprozesses für Knotengruppen werden laufende Instanzen ersetzt. Beachten Sie, dass dadurch alle aktiven Benutzersitzungen oder Prozesse auf der Instanz unterbrochen werden. Laufende oder in der Warteschlange befindliche Slurm-Jobs sind davon nicht betroffen. Weitere Informationen finden Sie unter Aktualisierung einer AWS PCS-Rechenknotengruppe.

Sie können auch die Startvorlage bearbeiten, die von Ihrer Compute-Knotengruppe verwendet wird. Sie müssen sie verwenden UpdateComputeNodeGroup , um die aktualisierte Startvorlage auf die Compute-Knotengruppe anzuwenden. Neue EC2-Instances, die in der Compute-Knotengruppe gestartet werden, verwenden die aktualisierte Startvorlage. Weitere Informationen finden Sie unter Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS.

Löschen einer AWS PCS-Compute-Knotengruppe für Anmeldeknoten

Sie können eine Anmeldeknotengruppe mithilfe des Mechanismus zum Löschen von Compute-Knotengruppen in AWS PCS aktualisieren. Laufende Instanzen werden im Rahmen des Löschens der Knotengruppe beendet. Bitte beachten Sie, dass dadurch alle aktiven Benutzersitzungen oder Prozesse auf der Instanz unterbrochen werden. Laufende oder in der Warteschlange befindliche Slurm-Jobs sind davon nicht betroffen. Weitere Informationen finden Sie unter Löschen einer Compute-Knotengruppe in AWS PCS.

Verwenden eigenständiger Instanzen als AWS PCS-Anmeldeknoten

Sie können unabhängige EC2-Instances einrichten, um mit dem Slurm-Scheduler eines AWS PCS-Clusters zu interagieren. Dies ist nützlich, um Anmeldeknoten, Workstations oder dedizierte Workflow-Management-Hosts zu erstellen, die mit AWS PCS-Clustern funktionieren, aber außerhalb des PCS-Managements betrieben werden. AWS Dazu muss jede eigenständige Instanz:

1. Eine kompatible Slurm-Softwareversion installiert haben.
2. In der Lage sein, eine Verbindung zum AWS Slurmctld-Endpunkt des PCS-Clusters herzustellen.
3. Sorgen Sie dafür, dass Slurm Auth und Cred Kiosk Daemon (`sackd`) ordnungsgemäß mit dem Endpunkt und dem Secret des PCS-Clusters konfiguriert sind. AWS Weitere Informationen finden Sie unter sackd in der Slurm-Dokumentation.

Dieses Tutorial hilft Ihnen bei der Konfiguration einer unabhängigen Instanz, die eine Verbindung zu einem AWS PCS-Cluster herstellt.

Inhalt

- Schritt 1 — Rufen Sie die Adresse und das Geheimnis für den AWS Ziel-PCS-Cluster ab
- Schritt 2 — Starten Sie eine EC2-Instanz
- Schritt 3 — Installieren Sie Slurm auf der Instanz
- Schritt 4 — Rufen Sie das Cluster-Geheimnis ab und speichern Sie es
- Schritt 5 — Konfigurieren Sie die Verbindung zum AWS PCS-Cluster
- Schritt 6 — (Optional) Testen Sie die Verbindung

Schritt 1 — Rufen Sie die Adresse und das Geheimnis für den AWS Ziel-PCS-Cluster ab

Rufen Sie mithilfe des folgenden Befehls Details zum AWS AWS CLI Ziel-PCS-Cluster ab. Nehmen Sie vor der Ausführung des Befehls die folgenden Ersetzungen vor:

- *region-code*Ersetzen Sie durch den AWS-Region Ort, an dem der Zielcluster ausgeführt wird.
- *cluster-ident*Ersetzen Sie durch den Namen oder die ID für den Zielcluster

```
aws pcs get-cluster --region region-code --cluster-identifier cluster-ident
```

Der Befehl gibt eine Ausgabe zurück, die diesem Beispiel ähnelt.

```
{
    "cluster": {
        "name": "get-started",
        "id": "pcs_123456abcd",
        "arn": "arn:aws:pcs:us-east-1:111122223333:cluster/pcs_123456abcd",
        "status": "ACTIVE",
        "createdAt": "2024-12-17T21:03:52+00:00",
        "modifiedAt": "2024-12-17T21:03:52+00:00",
        "scheduler": {
            "type": "SLURM",
            "version": "25.11"
        },
        "size": "SMALL",
        "slurmConfiguration": {
            "authKey": {
                "secretArn": "arn:aws:secretsmanager:us-east-1:111122223333:secret:pcs!
slurm-secret-pcs_123456abcd-a12ABC",
                "secretVersion": "ef232370-d3e7-434c-9a87-ec35c1987f75"
            }
        },
        "networking": {
            "subnetIds": [
                "subnet-0123456789abcdef0"
            ],
            "securityGroupIds": [
                "sg-0123456789abcdef0"
            ]
```

```
        },
        "endpoints": [
            {
                "type": "SLURMCTLD",
                "privateIpAddress": "10.3.149.220",
                "port": "6817"
            }
        ]
    }
}
```

In diesem Beispiel hat der Cluster-Slurm-Controller-Endpunkt die IP-Adresse `10.3.149.220` und er läuft auf dem Port`6817`. Der `secretArn` wird in späteren Schritten verwendet, um das Clustergeheimnis abzurufen. Die IP-Adresse und der Port werden in späteren Schritten zur Konfiguration des `sackd` Dienstes verwendet.

Schritt 2 — Starten Sie eine EC2-Instanz

Starten Sie EC2-Instances wie folgt:

1. Öffnen Sie die Amazon EC2-Konsole.
2. Wählen Sie im Navigationsbereich Instances und dann Instances starten aus, um den Launch Instance Wizard zu öffnen.
3. (Optional) Geben Sie im Abschnitt Name und Tags einen Namen für die Instance ein, z. B. `PCS-LoginNode` Der Name wird der Instance als Ressourcen-Tag (`Name=PCS-LoginNode`) zugewiesen.
4. Wählen Sie im Abschnitt Anwendungs- und Betriebssystemimages ein AMI für eines der von AWS PCS unterstützten Betriebssysteme aus. Weitere Informationen finden Sie unter Unterstützte Betriebssysteme.
5. Wählen Sie im Bereich Instance-Typ einen unterstützten Instance-Typ aus. Weitere Informationen finden Sie unter Unterstützte Instance-Typen.
6. Wählen Sie im Abschnitt key pair das SSH-Schlüsselpaar aus, das für die Instance verwendet werden soll.
7. Gehen Sie im Abschnitt Netzwerkeinstellungen wie folgt vor:
    - Wählen Sie Bearbeiten aus.
        i. Wählen Sie die VPC Ihres AWS PCS-Clusters aus.

ii. Für Firewall (Sicherheitsgruppen) wählen Sie Eine vorhandene Sicherheitsgruppe auswählen aus.

A. Wählen Sie eine Sicherheitsgruppe aus, die den Datenverkehr zwischen der Instanz und dem Slurm-Controller des AWS Ziel-PCS-Clusters zulässt. Weitere Informationen finden Sie unter Anforderungen und Überlegungen zur Sicherheitsgruppe.

B. (Optional) Wählen Sie eine Sicherheitsgruppe aus, die eingehenden SSH-Zugriff auf Ihre Instance ermöglicht.

8. Konfigurieren Sie im Bereich Speicher die Speichervolumes nach Bedarf. Stellen Sie sicher, dass ausreichend Speicherplatz für die Installation von Anwendungen und Bibliotheken konfiguriert ist, um Ihren Anwendungsfall zu unterstützen.
9. Wählen Sie unter Erweitert eine IAM-Rolle aus, die den Zugriff auf das Clustergeheimnis ermöglicht. Weitere Informationen finden Sie unter Hol dir das Geheimnis des Slurm-Clusters.
10. Wählen Sie im Übersichtsbereich die Option Launch instance aus.

Schritt 3 — Installieren Sie Slurm auf der Instanz

Wenn die Instanz gestartet wurde und aktiv wird, stellen Sie über Ihren bevorzugten Mechanismus eine Verbindung zu ihr her. Verwenden Sie das von bereitgestellte Slurm-Installationsprogramm AWS , um Slurm auf der Instanz zu installieren. Weitere Informationen finden Sie unter Slurm-Installationsprogramm.

Laden Sie das Slurm-Installationsprogramm herunter, dekomprimieren Sie es und verwenden Sie das `installer.sh` Skript, um Slurm zu installieren. Weitere Informationen finden Sie unter Schritt 3 — Slurm installieren.

Schritt 4 — Rufen Sie das Cluster-Geheimnis ab und speichern Sie es

Diese Anweisungen erfordern die AWS CLI. Weitere Informationen finden Sie unter Installation oder Aktualisierung auf die neueste Version von AWS CLI im AWS Command Line Interface Benutzerhandbuch für Version 2.

Speichern Sie das Clustergeheimnis mit den folgenden Befehlen.

- Erstellen Sie das Konfigurationsverzeichnis für Slurm.

```
sudo mkdir -p /etc/slurm
```

```
sudo chmod 0755 /etc/slurm
```

> **Note**
>
> Durch die Einstellung von Verzeichnisberechtigungen `0755` wird sichergestellt, dass der `slurm` Benutzer das Verzeichnis durchqueren kann, um auf die Schlüsseldatei zuzugreifen. Einige Systeme verfügen möglicherweise über eine restriktive Umask, die standardmäßig Verzeichnisse mit restriktiveren Berechtigungen erstellt.

- Rufen Sie das Clustergeheimnis ab, dekodieren Sie es und speichern Sie es. Bevor Sie diesen Befehl ausführen, *`region-code`* ersetzen Sie ihn durch die Region, in der der Zielcluster ausgeführt wird, und *`secret-arn`* durch den in Schritt 1 `secretArn` abgerufenen Wert.

```
aws secretsmanager get-secret-value \
 --region region-code \
 --secret-id 'secret-arn' \
 --version-stage AWSCURRENT \
 --query 'SecretString' \
 --output text | base64 -d | sudo tee /etc/slurm/slurm.key
```

> **Warning**
>
> In einer Mehrbenutzerumgebung kann möglicherweise jeder Benutzer mit Zugriff auf die Instance das Clustergeheimnis abrufen, wenn er auf den Instance-Metadatendienst (IMDS) zugreifen kann. Dies wiederum könnte es ihnen ermöglichen, sich als andere Benutzer auszugeben. Erwägen Sie, den Zugriff auf IMDS nur auf Root- oder Administratorbenutzer zu beschränken. Erwägen Sie alternativ, einen anderen Mechanismus zu verwenden, der sich nicht auf das Instanzprofil stützt, um den geheimen Schlüssel abzurufen und zu konfigurieren.

- Legen Sie den Besitz und die Berechtigungen für die Slurm-Schlüsseldatei fest.

```
sudo chmod 0600 /etc/slurm/slurm.key
sudo chown slurm:slurm /etc/slurm/slurm.key
```

> **Note**
>
> Der Slurm-Schlüssel muss dem Benutzer und der Gruppe gehören, unter denen der `sackd` Dienst ausgeführt wird.

Schritt 5 — Konfigurieren Sie die Verbindung zum AWS PCS-Cluster

Gehen Sie wie folgt vor, um eine Verbindung zum AWS PCS-Cluster herzustellen, indem Sie ihn `sackd` als Systemdienst starten.

> **Note**
>
> Wenn Sie Slurm 25.05 oder höher verwenden, können Sie stattdessen ein Skript verwenden, um Ihren Anmeldeknoten so einzurichten, dass er eine Verbindung zu mehreren Clustern herstellt. Weitere Informationen finden Sie unter Verbinden eines eigenständigen Anmeldeknotens mit mehreren Clustern in AWS STCK.

1. Richten Sie die Umgebungsdatei für den `sackd` Dienst mit dem folgenden Befehl ein. Bevor Sie den Befehl ausführen, ersetzen Sie *`ip-address`* und *`port`* durch die in Schritt 1 von den Endpunkten abgerufenen Werte.

   ```
   sudo echo "SACKD_OPTIONS='--conf-server=ip-address:port'" > /etc/sysconfig/sackd
   ```

2. Erstellen Sie eine `systemd` Servicedatei für die Verwaltung des `sackd` Prozesses.

   ```
   sudo cat << EOF > /etc/systemd/system/sackd.service
   [Unit]
   Description=Slurm auth and cred kiosk daemon
   After=network-online.target remote-fs.target
   Wants=network-online.target
   ConditionPathExists=/etc/sysconfig/sackd

   [Service]
   Type=notify
   EnvironmentFile=/etc/sysconfig/sackd
   User=slurm
   Group=slurm
   ```

```
RuntimeDirectory=slurm
RuntimeDirectoryMode=0755
ExecStart=/opt/aws/pcs/scheduler/slurm-25.11/sbin/sackd --systemd \$SACKD_OPTIONS
ExecReload=/bin/kill -HUP \$MAINPID
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF
```

3. Legen Sie den Besitz der `sackd` Servicedatei fest.

```
sudo chown root:root /etc/systemd/system/sackd.service && \
    sudo chmod 0644 /etc/systemd/system/sackd.service
```

4. Aktivieren Sie den `sackd` Dienst.

```
sudo systemctl daemon-reload && sudo systemctl enable sackd
```

5. Starten Sie den Service `sackd`.

```
sudo systemctl start sackd
```

Schritt 6 — (Optional) Testen Sie die Verbindung

Vergewissern Sie sich, dass der `sackd` Dienst läuft. Beispiel für eine Ausgabe folgt. Wenn es Fehler gibt, werden sie normalerweise hier angezeigt.

```
[root@ip-10-3-27-112 ~]# systemctl status sackd
[x] sackd.service - Slurm auth and cred kiosk daemon
   Loaded: loaded (/etc/systemd/system/sackd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-12-17 16:34:55 UTC; 8s ago
 Main PID: 9985 (sackd)
   CGroup: /system.slice/sackd.service
           ##9985 /opt/aws/pcs/scheduler/slurm-25.11/sbin/sackd --systemd --conf-
server=10.3.149.220:6817
```

```
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Starting Slurm auth and cred
 kiosk daemon...
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal systemd[1]: Started Slurm auth and cred
 kiosk daemon.
Dec 17 16:34:55 ip-10-3-27-112.ec2.internal sackd[9985]: sackd: running
```

Vergewissern Sie sich, dass die Verbindungen zum Cluster funktionieren, indem Sie Slurm-Client-Befehle wie `sinfo` und `squeue` verwenden. Hier ist eine Beispielausgabe von`sinfo`.

```
[root@ip-10-3-27-112 ~]# /opt/aws/pcs/scheduler/slurm-25.05/bin/sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
all up infinite 4 idle~ compute-[1-4]
```

Sie sollten auch Jobs einreichen können. Ein Befehl, der diesem Beispiel ähnelt, würde beispielsweise einen interaktiven Job auf einem Knoten im Cluster starten.

```
/opt/aws/pcs/scheduler/slurm-25.05/bin/srun --nodes=1 -p all --pty bash -i
```

Verbinden eines eigenständigen Anmeldeknotens mit mehreren Clustern in AWS STCK

Das `pcs-multi-cluster-login-configure.sh` Skript bietet eine automatisierte Möglichkeit, mehrere `sackd` Slurm-Daemons auf einem einzigen eigenständigen Anmeldeknoten zu konfigurieren. Es ermöglicht dem Anmeldeknoten, mit mehreren Clustern zu kommunizieren. Das Skript automatisiert die folgenden Operationen:

- Verwendet AWS PCS-API-Aktionen, um Cluster-Informationen abzurufen
- Fordert zur Eingabe des Base64-kodierten Slurm-Authentifizierungsschlüssels auf
- Erstellt eine Slurm-JWKS-Datei mit dem Cluster-Authentifizierungsschlüssel
- Konfiguriert den `sackd` Dienst mit Cluster-Endpunkten und -Ports
- Erstellt eine `systemd` Servicedatei für einen clusterspezifischen Daemon `sackd`
- Generiert ein Aktivierungsskript für die Einrichtung der Cluster-Umgebung
- Aktiviert und startet den `sackd` Dienst

> **Note**
>
> Dieses Skript benötigt Slurm Version 25.05 oder höher.

> **Note**
>
> Für Slurm 25.11 oder höher können Sie anstelle der Umgebungsvariablen `sackd --jwks-file <path>` und `sackd --key-file <path>` zur Angabe von Authentifizierungsschlüsselpfaden verwenden. `SLURM_SACK_JWKS` Der `SLURM_SACK_JWKS` Ansatz wird aus Gründen der Abwärtskompatibilität mit Slurm 25.05-Clustern weiterhin unterstützt.

Slurm muss bereits auf der Instanz installiert sein (entspricht Schritt 3 im manuellen Prozess). Die Instanz muss in der Lage sein, die Endpunkte des Zielclusters zu erreichen. Das Skript führt die entsprechenden Operationen aus Schritt 4 und Schritt 5 im manuellen Konfigurationsprozess aus. Es ruft automatisch die Cluster-Informationen ab, konfiguriert den `sackd` Dienst, erstellt die erforderlichen `systemd` Servicedateien und erstellt ein Aktivierungsskript, mit dem Benutzer ihre Shell-Umgebung für die Cluster-Interaktion konfigurieren können.

Standardmäßig ruft das Skript die AWS PCS-API auf, um die Cluster-Details nachzuschlagen. Um diesen Aufruf zu überspringen, geben Sie die Details direkt in der Befehlszeile mit den `--endpoints` Optionen `--cluster-name--cluster-id`,`--slurm-version`, und ein. Wenn Sie alle vier angeben, führt das Skript keinen AWS PCS-API-Aufruf durch. Dies ist nützlich für einen Anmeldeknoten, der keine Route zum AWS PCS-API-Endpunkt oder keine AWS Anmeldeinformationen hat. Die `--fetch-secret` Option ist eine Ausnahme: Sie ruft immer die AWS PCS-API auf, um den ARN des Secrets zu lesen.

Sie führen das Skript einmal für jeden Cluster aus, sodass ein einzelner eigenständiger Anmeldeknoten eine Verbindung zu mehreren Clustern gleichzeitig herstellen kann. Der Anmeldeknoten muss in der Lage sein, den `slurmctld` Controller jedes Clusters zu erreichen. Wenn zwei Cluster denselben Namen haben, verwenden Sie die `--alias` Option, um jedem Cluster eine eigene lokale ID auf dem Anmeldeknoten zuzuweisen. Verwenden Sie für Cluster in verschiedenen Clustern die `--region` Option AWS-Regionen, um die Region jedes Clusters anzugeben. Der Anmeldeknoten muss über einen privaten Netzwerkpfad zum `slurmctld` Controller dieser Region verfügen. Sie können beispielsweise eine VPC-Peering-Verbindung oder ein Transit-Gateway verwenden.

Topics

- Voraussetzungen für das Konfigurationsskript für den AWS PCS-Multi-Cluster-Login-Knoten
- AWS Konfigurationsskriptcode für den PCS-Multi-Cluster-Anmeldeknoten
- Verwenden des Konfigurationsskripts für den AWS PCS-Multi-Cluster-Anmeldeknoten

Voraussetzungen für das Konfigurationsskript für den AWS PCS-Multi-Cluster-Login-Knoten

Systemanforderungen

- Linux-Betriebssystem mit Unterstützung `systemd`
- Root-Rechte für die Systemkonfiguration

Erforderliche Befehle und Pakete

- `bash`— Shell-Interpreter (Version 4.0+)
- `curl`— Für den Abruf von AWS IMDS v2-Metadaten
- `jq`— JSON-Prozessor zum Analysieren von API-Antworten AWS
- `aws`— AWS CLI v2 zur Ausführung von AWS PCS-API-Aktionen und für den Secrets Manager Manager-Zugriff
- `systemctl`— `systemd` Servicemanagement
- `find`— Suchprogramm für das Dateisystem
- `grep`— Abgleich von Textmustern
- `sed`— Stream-Editor zur Textmanipulation
- `sort`— Hilfsprogramm zur Textsortierung
- `tail`— Zeigt die letzten Zeilen einer Datei an
- `mkdir`— Erstellung eines Verzeichnisses
- `chmod`— Ändert die Dateiberechtigungen
- `chown`— Ändert den Dateibesitz
- `ldconfig`— Dynamische Linker-Konfiguration

AWS Anforderungen

- Ein AWS PCS-Cluster, auf dem Slurm Version 25.05 oder höher ausgeführt wird
- AWS konfigurierte Anmeldeinformationen (über eine IAM-Rolle, eine Anmeldeinformationsdatei oder Umgebungsvariablen)
- Berechtigungen für:
  - `pcs:GetCluster`
  - `secretsmanager:GetSecretValue`(wenn Sie ein alternatives Geheimnis verwenden)

Systembenutzer und -gruppen

- Der `slurm` Benutzer und die Gruppe müssen auf dem System vorhanden sein

Slurm-Installation

- Slurm muss am selben Ort wie die AWS PCS Slurm-Installationspakete installiert werden:

```
/opt/aws/pcs/scheduler/slurm-version
```

AWS Konfigurationsskriptcode für den PCS-Multi-Cluster-Anmeldeknoten

Speichern Sie den folgenden Quellcode in einer Datei mit dem folgenden Namen:

```
pcs-multi-cluster-login-configure.sh
```

Quellcode des Skripts

```
#!/bin/bash
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# AWS PCS Multi-Cluster Standalone Login Node Configuration Script
#
# This script configures AWS Parallel Computing Service (PCS) multi-cluster stand alone
 login nodes
# by setting up the Slurm authentication and credential kiosk daemon (sackd)
# for connecting to remote PCS clusters. Run once for each cluster; clusters might
# be in different Regions.
```

```
#
#   ./pcs-multi-cluster-login-configure.sh --cluster-identifier <id> --region us-west-2
 --fetch-secret
#   ./pcs-multi-cluster-login-configure.sh --cluster-identifier <id> --region us-east-1
 --fetch-secret
#
# Prerequisites:
# - AWS CLI configured with appropriate permissions
# - Slurm version 25.05 or later
# - Root privileges for system configuration
# - Network connectivity to the AWS PCS control-plane API, used by this script to
#   look up the cluster
# - Network connectivity from this host to the cluster's slurmctld endpoint (TCP
#   6817 by default), used by sackd at runtime. This is a separate network path
#   from the preceding API path, and reaching one does not imply reaching the
#   other. The API might be served over a public endpoint, whereas slurmctld is
#   private to the cluster's VPC. For a cluster in another VPC or Region,
#   slurmctld needs peering or a transit gateway plus the matching routes and
#   security group rules.
# - For --fetch-secret: secretsmanager:GetSecretValue permission on the cluster's auth
 key
#
# Two clusters in different Regions can share a name. Use --alias to give each one a
# distinct local identifier, so their services and files do not collide:
#
#   ./pcs-multi-cluster-login-configure.sh --cluster-identifier <id> --region us-west-2
 --alias prod-west
#   ./pcs-multi-cluster-login-configure.sh --cluster-identifier <id> --region us-east-1
 --alias prod-east
#
# --cluster-name, --cluster-id, --slurm-version and --endpoints override what the AWS
# PCS API reports. Supply all four and the script makes no API call, which suits a host
# with no route to the API. Supply some and the call fills in the rest. --fetch-secret
# always calls the API to read the secret ARN, so it is an exception.
#
# The script checks the Slurm build before it changes anything. It does not probe
# controller reachability, because sackd itself is the authority on that. The script
# therefore starts sackd last and reports any failure to start with a pointer to
# its logs.


set -eo pipefail

# Function to display usage
```

```
usage() {
    echo "Usage: $0 --cluster-identifier <cluster-identifier> [--region <aws-region>]
 [--fetch-secret] [--endpoint-url <endpoint-url>] [--alias <name>]"
    echo "       $0 --cluster-identifier <id> --cluster-name <name> --cluster-id <id>
 --slurm-version <ver> --endpoints <host:port>"
    echo "       $0 -h|--help"
}

# Function to display help
help() {
    echo "AWS PCS Multi-Cluster Standalone Login Node Configuration Script"
    echo "================================================="
    echo
    echo "This script configures multi-cluster standalone login node for AWS Parallel
 Computing Service (PCS)"
    echo "by setting up the Slurm authentication and credential kiosk daemon (sackd)."
    echo
    echo "Supports clusters across multiple AWS Regions. Run the script once for each
 cluster."
    echo
    usage
    echo
    echo "Options:"
    echo "  --cluster-identifier <id>    AWS PCS cluster identifier (required)"
    echo "  --region <region>            AWS Region of the cluster (optional, defaults
 to instance Region via IMDS)"
    echo "  --fetch-secret               Automatically retrieve the Slurm auth key from
 AWS Secrets Manager"
    echo "                               (requires secretsmanager:GetSecretValue
 permission on the cluster's auth key)"
    echo "  --endpoint-url <url>         Custom PCS endpoint URL (optional, applies
 only when"
    echo "                               the script calls the API)"
    echo "  --alias <name>               Local identifier for this cluster (default:
 the cluster's"
    echo "                               own name). Use it when two clusters in
 different Regions"
    echo "                               share a name. Letters, digits, dot,
 underscore, hyphen."
    echo "  -h, --help                   Show this help message"
    echo
    echo "Overrides for what the AWS PCS API reports. Supply all four and the script
 makes no"
```

```
    echo "API call; supply some and the call fills in the rest. --fetch-secret reads
 the secret"
    echo "ARN from that call, so it always requires one:"
    echo "  --cluster-name <name>        Cluster name"
    echo "  --cluster-id <id>            Cluster ID, used as the Slurm JWKS key ID"
    echo "  --slurm-version <version>    Slurm version, selects /opt/aws/pcs/scheduler/
slurm-<version>"
    echo "  --endpoints <host:port>      slurmctld endpoints, comma separated"
    echo
    echo "Examples:"
    echo "  # Configure for a cluster in the same Region as this instance"
    echo "  $0 --cluster-identifier my-pcs-cluster"
    echo
    echo "  # Configure for clusters in multiple Regions"
    echo "  $0 --cluster-identifier cluster-west --region us-west-2"
    echo "  $0 --cluster-identifier cluster-east --region us-east-1"
    echo
    echo "  # Auto-fetch the auth key from Secrets Manager (no manual paste needed)"
    echo "  $0 --cluster-identifier my-pcs-cluster --fetch-secret"
    echo
    echo "  # Two clusters that share a name, told apart by alias"
    echo "  $0 --cluster-identifier cluster-west --region us-west-2 --alias prod-west"
    echo "  $0 --cluster-identifier cluster-east --region us-east-1 --alias prod-east"
    echo
    echo "  # No API call: every cluster detail supplied on the command line"
    echo "  $0 --cluster-identifier pcs_abc123 --cluster-name pcs-prod \\"
    echo "     --cluster-id pcs_abc123 --slurm-version 25.05 \\"
    echo "     --endpoints 10.0.1.157:6817 --region us-west-2"
    echo
    echo "Note: This script requires root privileges and Slurm version 25.05 or later."
    echo "      Cross-Region clusters require VPC peering or a transit gateway to the
 cluster's VPC."
}

# Function to retrieve authentication key
get_auth_key() {
    if [ "$ALTERNATE_SECRET_RETRIEVAL" = "true" ]; then
        echo "Retrieving authentication key from AWS Secrets Manager..." >&2
        local auth_key_arn=$(echo "$CLUSTER_INFO" | jq -r
 '.cluster.slurmConfiguration.authKey.secretArn')
        local auth_key_version=$(echo "$CLUSTER_INFO" | jq -r
 '.cluster.slurmConfiguration.authKey.secretVersion')

        if [ "$auth_key_arn" = "null" ] || [ "$auth_key_version" = "null" ]; then
```

```
            echo "Error: Auth key information not found in cluster configuration" >&2
            exit 1
        fi

        if ! aws secretsmanager get-secret-value --secret-id "$auth_key_arn" --version-
id "$auth_key_version" --query SecretString --output text --region "$REGION" 2>/dev/
null; then
            echo "Error: Failed to retrieve auth key from Secrets Manager" >&2
            exit 1
        fi
    else
        echo "Please enter the base64-encoded Slurm authentication key:" >&2
        echo -n "Base64 of the Slurm secret key: " >&2
        local key
        read -rs key
        echo >&2
        echo "$key"
    fi
}

# Function to get the instance Region from IMDS
get_imds_region() {
    # Try IPv6 IMDS endpoint first (fd00:ec2::254) with fast timeout (1s connect, 2s
 total)
    # If IPv6 fails, fall back to IPv4 IMDS endpoint (169.254.169.254)
    local imds_endpoint="http://[fd00:ec2::254]"
    local token
    if ! token=$(curl -s -X PUT "${imds_endpoint}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --connect-timeout 1 --max-time 2 2>/dev/null); then
        imds_endpoint="http://169.254.169.254"
        if ! token=$(curl -s -X PUT "${imds_endpoint}/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600" --max-time 5); then
            echo "Error: Failed to retrieve IMDS token. Ensure this script is running
 on an EC2 instance." >&2
            exit 1
        fi
    fi

    local doc region
    # Capture the document first so a curl or timeout failure is reported here instead
 of
    # aborting silently under set -e. jq then prints the literal "null" (exit 0) when
    # .region is absent, so the extracted value is also checked explicitly.
```

```
    if ! doc=$(curl -s -H "X-aws-ec2-metadata-token: $token" "${imds_endpoint}/latest/
dynamic/instance-identity/document" --max-time 5); then
        echo "Error: Failed to retrieve instance identity document from IMDS" >&2
        exit 1
    fi
    region=$(echo "$doc" | jq -r '.region')
    if [ -z "$region" ] || [ "$region" = "null" ]; then
        echo "Error: Failed to retrieve AWS Region from instance metadata" >&2
        exit 1
    fi
    echo "$region"
}

# Function to get next available SACKD port
get_next_sackd_port() {
    local exclude_file="$1"
    local port=6918
    local used_ports=()

    # Get all currently used SACKD ports into an array
    while IFS= read -r line; do
        used_ports+=("$line")
    done < <(find /etc/sysconfig -name "sackd-pcs-*" ! -path "$exclude_file" \
             -exec grep SACKD_PORT= '{}' ';' 2>/dev/null | \
             sed 's/.*SACKD_PORT=//' | sort -n)

    # Loop through used ports to find first available port
    for used_port in "${used_ports[@]}"; do
        if [ "$port" -lt "$used_port" ]; then
            break
        elif [ "$port" -eq "$used_port" ]; then
            ((port++))
        fi
    done

    echo "$port"
}

# Function to configure cluster
configure_cluster() {
    # The mode is explicit because a restrictive umask would otherwise leave this
    # directory untraversable by the slurm user, which sackd runs as.
    install -d -m 0755 /etc/slurm
    SLURM_JWKS_FILE="/etc/slurm/slurm-${CLUSTER_LOCAL}.jwks"
```

```
    # Create the file with restrictive permissions before writing the secret, so the
    # base64 Slurm key is never briefly readable by other users. The truncating
 redirect
    # below preserves this mode and ownership, so no later chmod/chown is needed.
    install -m 0600 -o slurm -g slurm /dev/null "${SLURM_JWKS_FILE}"
    echo '{"keys":
[{"alg":"HS256","kty":"oct","kid":"key-'"${CLUSTER_ID}"'","k":"'"${BASE64_SLURM_KEY}"'"}]}'
 | jq -c '.' > "${SLURM_JWKS_FILE}"

    # main() sets and validates SLURM_INSTALL_PATH before it calls this function.

    SACKD_RUNTIME_DIRECTORY="/run/slurm-${CLUSTER_LOCAL}"
    mkdir -p "${SACKD_RUNTIME_DIRECTORY}"
    chown slurm:slurm "${SACKD_RUNTIME_DIRECTORY}"

    mkdir -p /etc/sysconfig
    SACKD_SERVICE_NAME="sackd-pcs-${CLUSTER_LOCAL}"
    SACKD_SERVICE_ENV="/etc/sysconfig/${SACKD_SERVICE_NAME}"
    SACKD_PORT=$(get_next_sackd_port "$SACKD_SERVICE_ENV")
    cat > "${SACKD_SERVICE_ENV}" << EOF
SACKD_OPTIONS='--conf-server=$ENDPOINTS'
SLURM_SACK_JWKS='$SLURM_JWKS_FILE'
RUNTIME_DIRECTORY='$SACKD_RUNTIME_DIRECTORY'
SACKD_PORT=$SACKD_PORT
EOF

    SACKD_SERVICE_PATH="/etc/systemd/system/${SACKD_SERVICE_NAME}.service"

    cat << EOF > "$SACKD_SERVICE_PATH"
[Unit]
Description=Slurm auth and cred kiosk daemon (${CLUSTER_LOCAL} - ${REGION})
After=network-online.target remote-fs.target
Wants=network-online.target
ConditionPathExists=${SACKD_SERVICE_ENV}

[Service]
Type=notify
EnvironmentFile=${SACKD_SERVICE_ENV}
User=slurm
Group=slurm
RuntimeDirectory=slurm-${CLUSTER_LOCAL}
RuntimeDirectoryMode=0755
ExecStart=${SLURM_INSTALL_PATH}/sbin/sackd --systemd \$SACKD_OPTIONS
ExecReload=/bin/kill -HUP \$MAINPID
```

```
KillMode=process
LimitNOFILE=131072
LimitMEMLOCK=infinity
LimitSTACK=infinity

[Install]
WantedBy=multi-user.target
EOF

    chown root:root "$SACKD_SERVICE_PATH"
    chmod 0644 "$SACKD_SERVICE_PATH"

    ACTIVATE_SCRIPT="activate-pcs-${CLUSTER_LOCAL}"
    cat > "$ACTIVATE_SCRIPT" << EOF
# Activate script for Slurm cluster ${CLUSTER_NAME} as ${CLUSTER_LOCAL} (Region:
 ${REGION})

# Add Slurm paths
export PATH="${SLURM_INSTALL_PATH}/bin:\$PATH"
export MANPATH="${SLURM_INSTALL_PATH}/share/man:\$MANPATH"
export LD_LIBRARY_PATH="${SLURM_INSTALL_PATH}/lib:\$LD_LIBRARY_PATH"
ldconfig

# Set Slurm configuration
export SLURM_CONF="/run/slurm-${CLUSTER_LOCAL}/conf/slurm.conf"

# Slurm clients otherwise derive the socket path from the cluster's own name, which
# an alias does not match.
export SLURM_SACK_SOCKET="/run/slurm-${CLUSTER_LOCAL}/sack.socket"

export PCS_CLUSTER_NAME="${CLUSTER_LOCAL}"
export PCS_CLUSTER_ACTUAL_NAME="${CLUSTER_NAME}"
export PCS_CLUSTER_IDENTIFIER="${CLUSTER_IDENTIFIER}"
export PCS_CLUSTER_ID="${CLUSTER_ID}"
export PCS_CLUSTER_REGION="${REGION}"

echo "Activated PCS cluster environment: ${CLUSTER_LOCAL} (${REGION})"

# Deactivate function
function deactivate-pcs-${CLUSTER_LOCAL}() {
    export PATH="\$(echo "\$PATH" | sed -e "s|${SLURM_INSTALL_PATH}/bin:||g" -e "s|:
${SLURM_INSTALL_PATH}/bin||g" -e "s|^${SLURM_INSTALL_PATH}/bin\$||")"
```

```
    export MANPATH="\$(echo "\$MANPATH" | sed -e "s|${SLURM_INSTALL_PATH}/share/man:||
g" -e "s|:${SLURM_INSTALL_PATH}/share/man||g" -e "s|^${SLURM_INSTALL_PATH}/share/man\
$||")"
    export LD_LIBRARY_PATH="\$(echo "\$LD_LIBRARY_PATH" | sed -e "s|
${SLURM_INSTALL_PATH}/lib:||g" -e "s|:${SLURM_INSTALL_PATH}/lib||g" -e "s|^
${SLURM_INSTALL_PATH}/lib\$||")"
    unset SLURM_CONF
    unset SLURM_SACK_SOCKET
    unset PCS_CLUSTER_NAME
    unset PCS_CLUSTER_ACTUAL_NAME
    unset PCS_CLUSTER_IDENTIFIER
    unset PCS_CLUSTER_ID
    unset PCS_CLUSTER_REGION
    unset -f deactivate-pcs-${CLUSTER_LOCAL}
    ldconfig
    echo "Deactivated PCS cluster environment: ${CLUSTER_LOCAL} (${REGION})"
}

export -f deactivate-pcs-${CLUSTER_LOCAL}

EOF

    # Start sackd last, after everything else is on disk. If it fails, the cluster is
    # still fully configured and the activate script exists, so the fix is to correct
    # whatever the logs point at and run 'systemctl restart'. There is no need to
    # re-run this script.
    systemctl daemon-reload && systemctl enable "$SACKD_SERVICE_NAME"
    if ! systemctl restart "$SACKD_SERVICE_NAME"; then
        cat >&2 << MSG

========================================
Configuration written, but sackd did not start
========================================
Service: ${SACKD_SERVICE_NAME}
Cluster: ${CLUSTER_NAME} as ${CLUSTER_LOCAL} (${REGION})
Endpoint: ${ENDPOINTS}

Check what went wrong:
  journalctl -xeu ${SACKD_SERVICE_NAME}
  systemctl status ${SACKD_SERVICE_NAME}

A start that hangs for ~90 seconds and then times out usually means this host
cannot open a TCP connection to the endpoint listed earlier. Look for:
```

```
  error: Failed to load configs from slurmctld. Retrying in 10 seconds.

For a cluster in another VPC or Region, confirm that VPC peering or transit
gateway routes exist on both sides and that security groups allow the traffic.
Being able to call the AWS PCS API does not imply the controller is reachable.

The configuration itself is complete. After you fix the cause, run:
  systemctl restart ${SACKD_SERVICE_NAME}

sackd might still be retrying in the background. To stop it:
  systemctl disable --now ${SACKD_SERVICE_NAME}

MSG
        exit 1
    fi
}

# Main function
main() {
    # Parse arguments
    CLUSTER_IDENTIFIER=""
    PCS_ENDPOINT_URL=""
    REGION=""
    ALTERNATE_SECRET_RETRIEVAL="false"
    # Local identifier for this cluster. Defaults to the cluster's own name once the
    # script knows it; set it explicitly to tell apart two clusters that share a name.
    CLUSTER_ALIAS=""
    CLUSTER_ALIAS_SET="false"
    # Overrides for what the AWS PCS API reports. All four set means the script skips
    # the call; any one unset means it makes the call and takes that value from it.
    ARG_CLUSTER_NAME=""
    ARG_CLUSTER_ID=""
    ARG_SLURM_VERSION=""
    ARG_ENDPOINTS=""

    while [ "$1" != "" ]; do
        case $1 in
            --cluster-identifier)
                shift
                CLUSTER_IDENTIFIER="$1"
                ;;
            --region)
                shift
                REGION="$1"
```

```
            ;;
        --fetch-secret)
            ALTERNATE_SECRET_RETRIEVAL="true"
            ;;
        --endpoint-url)
            shift
            PCS_ENDPOINT_URL="--endpoint-url $1"
            ;;
        --alias)
            shift
            CLUSTER_ALIAS="$1"
            CLUSTER_ALIAS_SET="true"
            ;;
        --cluster-name)
            shift
            ARG_CLUSTER_NAME="$1"
            ;;
        --cluster-id)
            shift
            ARG_CLUSTER_ID="$1"
            ;;
        --slurm-version)
            shift
            ARG_SLURM_VERSION="$1"
            ;;
        --endpoints)
            shift
            ARG_ENDPOINTS="$1"
            ;;
        -h|--help)
            help
            exit 0
            ;;
        *)
            echo "Invalid argument: $1" >&2
            usage >&2
            exit 1
            ;;
    esac
    shift
done

# Validate required arguments
if [ -z "$CLUSTER_IDENTIFIER" ]; then
```

```
        echo "Error: --cluster-identifier is required" >&2
        usage >&2
        exit 1
    fi

    # The alias becomes a filename, a systemd unit name and a bash function name.
    if [ "$CLUSTER_ALIAS_SET" = "true" ]; then
        case "$CLUSTER_ALIAS" in
            *[!A-Za-z0-9._-]*|"")
                echo "Error: --alias accepts only letters, digits, dot, underscore and
hyphen" >&2
                exit 1
                ;;
        esac
    fi

    # Validate running as root
    if [ "$EUID" -ne 0 ]; then
        echo "Error: This script must be run as root" >&2
        exit 1
    fi

    # Validate required commands are available
    for cmd in aws jq curl; do
        if ! command -v "$cmd" &> /dev/null; then
            echo "Error: Required command '$cmd' not found" >&2
            exit 1
        fi
    done

    # Get Region: use --region if provided, otherwise detect from IMDS
    if [ -z "$REGION" ]; then
        echo "No --region specified, detecting from instance metadata..."
        REGION=$(get_imds_region)
    fi
    echo "Using AWS Region: $REGION"

    # The API call answers whatever the command line did not, and it is the only source
    # of the secret ARN that --fetch-secret needs, so the script skips it only when
    # nothing remains to ask for.
    if [ -n "$ARG_CLUSTER_NAME" ] && [ -n "$ARG_CLUSTER_ID" ] &&
       [ -n "$ARG_SLURM_VERSION" ] && [ -n "$ARG_ENDPOINTS" ] &&
       [ "$ALTERNATE_SECRET_RETRIEVAL" = "false" ]; then
        echo "Every cluster detail was supplied; skipping the AWS PCS API call"
```

```
    else
        # Retrieve cluster information from AWS PCS
        echo "Retrieving cluster information for: $CLUSTER_IDENTIFIER (Region:
 $REGION)"
        # shellcheck disable=SC2086
        if ! CLUSTER_INFO=$(aws pcs get-cluster --region "$REGION" --cluster-identifier
 "$CLUSTER_IDENTIFIER" $PCS_ENDPOINT_URL 2>/dev/null); then
            echo "Error: Failed to retrieve cluster information. Check cluster
 identifier, Region, and AWS permissions." >&2
            exit 1
        fi

        CLUSTER_ID=$(echo "$CLUSTER_INFO" | jq -r '.cluster.id')
        CLUSTER_NAME="$(echo "$CLUSTER_INFO" | jq -r '.cluster.name')"
        SLURM_VERSION=$(echo "$CLUSTER_INFO" | jq -r '.cluster.scheduler.version')
        SLURM_VERSION=${SLURM_VERSION#Slurm_}
        ENDPOINTS=$(echo "$CLUSTER_INFO" | jq -r '.cluster.endpoints[] | select(.type
 == "SLURMCTLD") | (if .privateIpAddress != "" then .privateIpAddress else "["
 + .ipv6Address + "]" end) + ":" + .port' | tr '\n' ',' | sed 's/,$//')
    fi

    # A command-line value wins over whatever the call returned. When the script skips
    # the call, all four are set and the fallbacks never apply.
    CLUSTER_ID="${ARG_CLUSTER_ID:-$CLUSTER_ID}"
    CLUSTER_NAME="${ARG_CLUSTER_NAME:-$CLUSTER_NAME}"
    SLURM_VERSION="${ARG_SLURM_VERSION:-$SLURM_VERSION}"
    ENDPOINTS="${ARG_ENDPOINTS:-$ENDPOINTS}"

    # Local identifier for every file, directory, unit and function the script creates.
    CLUSTER_LOCAL="${CLUSTER_ALIAS:-$CLUSTER_NAME}"
    # CLUSTER_LOCAL becomes a filename, a systemd unit name and a bash function name,
 so it
    # must be validated whether it came from --alias or from an unvalidated --cluster-
name.
    case "$CLUSTER_LOCAL" in
        *[!A-Za-z0-9._-]*|"")
            echo "Error: cluster local name '$CLUSTER_LOCAL' must contain only letters,
 digits, dot, underscore and hyphen" >&2
            exit 1
            ;;
    esac

    # Check if Slurm version is >= 25.05
    # shellcheck disable=SC2072
```

```
    if [[ "$SLURM_VERSION" < "25.05" ]]; then
        echo "Error: This script requires Slurm version 25.05 or later. Found version:
 $SLURM_VERSION" >&2
        exit 1
    fi

    # The following Slurm paths come from the cluster's version rather than from
    # local discovery, so confirm that build is actually present before writing
    # anything. Without this guard, the script writes the JWKS, env file, and unit,
    # then dies on 'systemctl restart' with a bare 203/EXEC, leaving secrets on
    # disk and an enabled service that can never start, with nothing to say why.
    SLURM_INSTALL_PATH="/opt/aws/pcs/scheduler/slurm-${SLURM_VERSION}"
    if [ ! -x "${SLURM_INSTALL_PATH}/sbin/sackd" ]; then
        echo "Error: cluster $CLUSTER_IDENTIFIER runs Slurm ${SLURM_VERSION}, but no
 such" >&2
        echo "build is installed on this host (expected ${SLURM_INSTALL_PATH}/sbin/
sackd)." >&2
        echo "Slurm versions available here:" >&2
        # The glob only ever matches PCS Slurm install directories, whose names are
        # version strings, so rewriting this with find gains nothing here. ls -1d
        # keeps the listing sorted and readable.
        # shellcheck disable=SC2012
        ls -1d /opt/aws/pcs/scheduler/slurm-* 2>/dev/null | sed 's|.*/slurm-|  |' >&2 \
            || echo "  (none found under /opt/aws/pcs/scheduler)" >&2
        exit 1
    fi

    if [ -z "$ENDPOINTS" ]; then
        echo "Error: cluster $CLUSTER_IDENTIFIER reports no SLURMCTLD endpoint." >&2
        echo "The cluster might still be provisioning. Check 'aws pcs get-cluster'."
 >&2
        exit 1
    fi

    # Get BASE64_SLURM_KEY
    BASE64_SLURM_KEY=$(get_auth_key)

    if [ -z "$BASE64_SLURM_KEY" ]; then
        echo "Error: base64 Slurm key cannot be empty" >&2
        exit 1
    fi

    configure_cluster
```

```
    # Final configuration summary
    echo "========================================"
    echo "Configuration completed successfully!"
    echo "========================================"
    echo "Cluster Name: $CLUSTER_NAME"
    echo "Local Alias: $CLUSTER_LOCAL"
    echo "Cluster ID: $CLUSTER_ID"
    echo "Region: $REGION"
    echo "Slurm Version: $SLURM_VERSION"
    echo "Service Name: $SACKD_SERVICE_NAME"
    echo "SACKD Port: $SACKD_PORT"
    echo
    echo "To activate this cluster environment, run:"
    echo "  source ./$ACTIVATE_SCRIPT"
    echo
    echo "To deactivate this cluster environment, run:"
    echo "  deactivate-pcs-${CLUSTER_LOCAL}"
    echo
    echo "To check service status:"
    echo "  systemctl status $SACKD_SERVICE_NAME"
    echo
    echo "To view service logs:"
    echo "  journalctl -u $SACKD_SERVICE_NAME -f"
}

# Exit if being sourced for testing
[[ "${BASH_SOURCE[0]}" != "${0}" ]] && return

# Execute main function
main "$@"
```

Verwenden des Konfigurationsskripts für den AWS PCS-Multi-Cluster-Anmeldeknoten

Ausführen des Skripts

So führen Sie das Konfigurationsskript aus:

1. Speichern Sie den Inhalt des Skripts in einer Datei mit dem Namen:

```
pcs-multi-cluster-login-configure.sh
```

2. Machen Sie es ausführbar:

```
chmod +x pcs-multi-cluster-login-configure.sh
```

3. Führen Sie das Skript aus:

```
./pcs-multi-cluster-login-configure.sh --cluster-identifier cluster-name
```

Umgebungen für Cluster-Interaktionen

Nach erfolgreicher Konfiguration generiert das Skript ein clusterspezifisches Aktivierungsskript im aktuellen Verzeichnis. Das Skript hat den Namen. `activate-pcs-cluster-name` Das Aktivierungsskript konfiguriert die erforderlichen Umgebungsvariablen und Pfade für die Interaktion mit dem Zielcluster.

Um eine Clusterumgebung zu aktivieren

- Verwenden Sie den `source` Befehl, um das Aktivierungsskript auszuführen

  ```
  source ./activate-pcs-cluster-name
  ```

  Example

  ```
  # Activate cluster environment for cluster 'my-cluster'
  source ./activate-pcs-my-cluster

  # Now you can use Slurm commands
  sinfo
  squeue
  sbatch my-job.sh
  ```

Was macht das Aktivierungsskript

- Legt die `SLURM_CONF` Umgebungsvariable so fest, dass sie auf die Konfiguration des Clusters verweist.
- Aktualisiert den`PATH`, sodass er die Slurm-Binärdateien des Clusters enthält.
- Konfiguriert andere notwendige Slurm-Umgebungsvariablen (,). `MANPATH` `LD_LIBRARY_PATH`

- Legt Variablen zur Identifizierung des AWS PCS-Clusters fest.
- Ermöglicht eine nahtlose Interaktion mit dem AWS PCS-Zielcluster.

Um eine Cluster-Umgebung zu deaktivieren

- Führen Sie den Deaktivierungsbefehl aus.

  ```
  deactivate-pcs-cluster-name
  ```

  Example

  ```
  # After activating a cluster
  source ./activate-pcs-my-cluster

  # Work with the cluster
  sinfo

  # Deactivate when done
  deactivate-pcs-my-cluster
  ```

Was macht der Deaktivierungsbefehl

- Stellt die ursprüngliche PATH Umgebungsvariable wieder her.
- Setzt clusterspezifische Slurm-Umgebungsvariablen zurück.
- Setzt die Shell-Umgebung in ihren Zustand vor der Aktivierung zurück.

> **Note**
>
> Die Aktivierung ist sitzungsspezifisch und muss in der Shell-Sitzung erfolgen, in der Sie mit dem Cluster interagieren möchten.

AWS PCS-Netzwerke

Ihr AWS PCS-Cluster wird in einer Amazon VPC erstellt. Dieses Kapitel enthält die folgenden Themen über Netzwerke für den Scheduler und die Knoten Ihres Clusters.

Abgesehen von der Auswahl eines Subnetzes, in dem Instances gestartet werden sollen, müssen Sie EC2 Startvorlagen verwenden, um das Netzwerk für AWS PCS-Compute-Knotengruppen zu konfigurieren. Weitere Informationen über Startvorlagen finden Sie unter Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS.

Themen

- AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen
- Erstellen Sie eine VPC für Ihre AWS PCS-Cluster
- Sicherheitsgruppen in AWS STK.
- Mehrere Netzwerkschnittstellen in AWS PCS
- Platzierungsgruppen für EC2-Instances in AWS PCS
- Verwenden des Elastic Fabric Adapter (EFA) mit AWS PCS

AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen

Wenn Sie einen AWS PCS-Cluster erstellen, geben Sie eine VPC als Subnetz in dieser VPC an. Dieses Thema bietet einen Überblick über die AWS PCS-spezifischen Anforderungen und Überlegungen für die VPC und die Subnetze, die Sie mit Ihrem Cluster verwenden. Wenn Sie keine VPC haben, die Sie mit AWS PCS verwenden können, können Sie eine mit einer AWS bereitgestellten CloudFormation Vorlage erstellen. Weitere Informationen zu VPCs finden Sie unter Virtual Private Clouds (VPC) im Amazon VPC-Benutzerhandbuch.

VPC-Anforderungen und -Überlegungen

Wenn Sie einen Cluster erstellen, muss die von Ihnen angegebene VPC die folgenden Anforderungen und Überlegungen erfüllen:

- Die VPC muss über eine ausreichende Anzahl von IP-Adressen für den Cluster, alle Knoten und andere Clusterressourcen verfügen, die Sie erstellen möchten. Weitere Informationen finden Sie unter IP-Adressierung für Ihre VPCs und Subnetze im Amazon VPC-Benutzerhandbuch.
- Wenn Ihr Cluster IPv6 verwendet:
  - Ordnen Sie Ihrer VPC einen IPv6-CIDR-Block zu. Weitere Informationen finden Sie unter VPC erstellen im Amazon-VPC-Benutzerhandbuch.

    > **Important**
    >
    > Sie können Ihre VPC zwar sowohl mit IPv4 als auch mit IPv6 konfigurieren, Sie können jedoch nur einen Netzwerktyp für Ihren Cluster auswählen.

  - Aktivieren Sie die automatische Zuweisung von IPv6-Adressen für Ihre Subnetze.
  - Weitere Informationen finden Sie unter:
    - IPv6 aktiviert AWS
    - Grundlegendes zur IPv6-Adressierung auf AWS und Entwicklung eines skalierbaren Adressierungsplans
- Die VPC muss über einen DNS-Hostnamen und eine Unterstützung für DNS-Auflösung verfügen. Andernfalls können Knoten den Kundencluster nicht registrieren. Weitere Infomationen finden Sie unter DNS-Attribute für Ihre VPC im Amazon VPC-Benutzerhandbuch.
- Für die VPC müssen möglicherweise VPC-Endpunkte verwendet werden AWS PrivateLink , um die PCS-API kontaktieren zu können. AWS Standardmäßig verwendet der AWS PCS-Agent den Dual-Stack-Endpunkt ohne FIPS. `pcs.region.api.aws` Wenn Sie ein benutzerdefiniertes DNS verwenden, müssen Sie manuell einen Eintrag für diesen Endpunkt hinzufügen. Weitere Informationen finden Sie unter Connect Ihrer VPC mit Services AWS PrivateLink im Amazon VPC-Benutzerhandbuch.

> **Important**
>
> AWS PCS unterstützt keine VPC mit dedizierter Instance-Tenancy. Die VPC, die Sie für AWS PCS verwenden, muss `default` Instance-Tenancy verwenden. Sie können die Instance-Tenancy für eine bestehende VPC ändern. Weitere Informationen finden Sie unter Ändern der Instance-Tenancy einer VPC im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Subnetz-Anforderungen und -Überlegungen

Wenn Sie einen Slurm-Cluster erstellen, erstellt AWS PCS ein Elastic Network Interface (ENI) in dem von Ihnen angegebenen Subnetz. Diese Netzwerkschnittstelle ermöglicht die Kommunikation zwischen dem Scheduler-Controller und der Kunden-VPC. Die Netzwerkschnittstelle ermöglicht es Slurm auch, mit den in Ihrem Konto bereitgestellten Komponenten zu kommunizieren. Sie können das Subnetz für einen Cluster nur zum Zeitpunkt der Erstellung angeben.

Subnetzanforderungen für Cluster

Das Subnetz, das Sie bei der Erstellung eines Clusters angeben, muss die folgenden Anforderungen erfüllen:

- Das Subnetz muss mindestens eine IP-Adresse haben, damit es von PCS verwendet werden AWS kann.
- Wenn Ihr Cluster IPv6 verwendet, müssen alle Subnetze in Ihrem Cluster IPv6 verwenden.

> **Important**
>
> Compute-Knotengruppen, die mit AWS PCS-Beispiel-AMIs und mehreren Netzwerkschnittstellen konfiguriert sind, funktionieren derzeit nicht, wenn die Subnetze nur für die Verwendung von IPv6 konfiguriert sind. Verwenden Sie stattdessen Dual-Stack-Subnetze (IPv4 und IPv6) oder Subnetze. IPv4-only Weitere Informationen finden Sie unter Verwenden von Amazon Machine Images (AMIs) als Beispiel mit AWS STCK..

- Das Subnetz kann sich nicht in AWS Outposts, oder einer lokalen Zone befinden. AWS Wavelength AWS
- Das Subnetz kann öffentlich oder privat sein. Wir empfehlen, dass Sie, wenn möglich, ein privates Subnetz angeben. Ein öffentliches Subnetz ist ein Subnetz mit einer Routing-Tabelle, die eine Route zu einem Internet-Gateway enthält. Ein privates Subnetz ist ein Subnetz mit einer Routing-Tabelle, das keine Route zu einem Internet-Gateway enthält.

Subnetzanforderungen für Knoten

Sie können Knoten und andere Clusterressourcen in dem Subnetz bereitstellen, das Sie bei der Erstellung Ihres AWS PCS-Clusters angeben, sowie in anderen Subnetzen in derselben VPC.

Jedes Subnetz, in dem Sie Knoten und Clusterressourcen bereitstellen, muss die folgenden Anforderungen erfüllen:

- Sie müssen sicherstellen, dass das Subnetz über genügend verfügbare IP-Adressen verfügt, um alle Knoten und Clusterressourcen bereitzustellen.
- Wenn Ihr Cluster IPv4 verwendet und Sie planen, Knoten in einem öffentlichen Subnetz bereitzustellen, muss dieses Subnetz automatisch öffentliche IPv4-Adressen zuweisen.

> **Note**
>
> Instances in einem öffentlichen Subnetz müssen eine Sicherheitsgruppe mit Regeln für eingehenden Datenverkehr verwenden, die Datenverkehr von öffentlichen IP-Adressen zulassen. Sofern Sie keine spezifischen Einschränkungen für Quelladressen haben, bedeutet dies eine IPv4-Quelladresse von 0,0.0. 0/0oder eine IPv6-Quelladresse von: :/0.

- Wenn es sich bei dem Subnetz, in dem Sie Knoten bereitstellen, um ein privates Subnetz handelt und dessen Routing-Tabelle keine Route zu einem Network Address Translation (NAT) -Gerät (IPv4) enthält, fügen Sie VPC-Endpunkte mithilfe AWS PrivateLink der Kunden-VPC hinzu. VPC-Endpunkte werden für alle AWS Dienste benötigt, mit denen die Knoten Kontakt aufnehmen. Der einzige erforderliche Endpunkt besteht darin, dass AWS PCS es dem Knoten ermöglicht, die `RegisterComputeNodeGroupInstance` API-Aktion aufzurufen. Weitere Informationen finden Sie RegisterComputeNodeGroupInstancein der AWS PCS-API-Referenz.
- Der Status eines öffentlichen oder privaten Subnetzes hat keinen Einfluss auf AWS PCS. Die erforderlichen Endpunkte müssen erreichbar sein.

Erstellen Sie eine VPC für Ihre AWS PCS-Cluster

Sie können eine Amazon Virtual Private Cloud (Amazon VPC) für Ihre Cluster erstellen AWS Dienst für parallele Datenverarbeitung (AWS STÜCK).

Verwenden Sie Amazon VPC, um VPC-Ressourcen in einem von Ihnen definierten virtuellen Netzwerk zu starten. Dieses virtuelle Netzwerk ist einem herkömmlichen Netzwerk, das Sie in Ihrem eigenen Rechenzentrum betreiben, sehr ähnlich. Es bietet jedoch die Vorzüge, die mit der Nutzung der skalierbaren Infrastruktur von Amazon Web Services einhergehen. Wir empfehlen, dass Sie den Amazon VPC-Service gründlich verstehen, bevor Sie VPC-Produktionscluster bereitstellen. Weitere Informationen finden Sie unter Was ist Amazon VPC? im visuellen Autorenmodus.Amazon VPC-Benutzerhandbuch.

Ein PCS-Cluster, Knoten und unterstützende Ressourcen (wie Dateisysteme und Verzeichnisdienste) werden in Ihrer Amazon VPC bereitgestellt. Wenn Sie eine bestehende Amazon VPC mit PCS verwenden möchten, muss diese die unter beschriebenen Anforderungen erfüllen AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen . In diesem Thema wird beschrieben, wie Sie eine VPC erstellen, die die PCS-Anforderungen erfüllt, mithilfe eines AWS—bereitgestellt CloudFormation Vorlage. Sobald Sie eine Vorlage bereitgestellt haben, können Sie sich die mit der Vorlage erstellten Ressourcen ansehen, um genau zu erfahren, welche Ressourcen sie erstellt hat und wie diese Ressourcen konfiguriert sind.

Voraussetzungen

Um eine Amazon VPC für PCS zu erstellen, benötigen Sie die erforderlichen IAM-Berechtigungen, um Amazon VPC-Ressourcen zu erstellen. Diese Ressourcen sind VPCs, Subnetze, Sicherheitsgruppen, Routing-Tabellen und Routen sowie Internet- und NAT-Gateways. Weitere Informationen finden Sie unter Erstellen Sie eine VPC mit einem öffentlichen Subnetz in der Amazon VPC-Benutzerhandbuch. Die vollständige Liste für Amazon EC2 finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EC2 in der Referenz zur Serviceautorisierung.

Erstellen Sie eine Amazon VPC

Erstellen Sie eine VPC, indem Sie die entsprechende URL für das kopieren und einfügen AWS-Region wo Sie PCS verwenden werden. Sie können auch das herunterladen CloudFormation Vorlage und laden Sie sie selbst hoch auf CloudFormation Konsole.

- USA Ost (Nord-Virginia) (us-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- USA Ost (Ohio) (us-east-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-east-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- USA West (Oregon) (us-west-2)

```
https://console.aws.amazon.com/cloudformation/home?region=us-west-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Afrika (Kapstadt) (af-south-1)

```
https://console.aws.amazon.com/cloudformation/home?region=af-south-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Mumbai) (ap-south-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-south-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Hyderabad) (ap-south-2)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-south-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Singapur) (ap-southeast-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Sydney) (ap-southeast-2)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Jakarta) (ap-southeast-3)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-3#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Melbourne) (ap-southeast-4)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-4#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Malaysia) (ap-southeast-5)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-5#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Neuseeland) (ap-southeast-6)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-southeast-6#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Tokio) (ap-northeast-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-northeast-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Seoul) (ap-northeast-2)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-northeast-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Asien-Pazifik (Osaka) (ap-northeast-3)

```
https://console.aws.amazon.com/cloudformation/home?region=ap-northeast-3#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Kanada (Zentral) (ca-central-1)

```
https://console.aws.amazon.com/cloudformation/home?region=ca-central-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (Frankfurt) (eu-central-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-central-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (Irland) (eu-west-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-west-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (London) (eu-west-2)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-west-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (Paris) (eu-west-3)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-west-3#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (Mailand) (eu-Süd-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-south-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (Spanien) (eu-Süd-2)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-south-2#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Europa (Stockholm) (eu-nord-1)

```
https://console.aws.amazon.com/cloudformation/home?region=eu-north-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Südamerika (São Paulo) (sa-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=sa-east-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- AWS GovCloud (US-East) (us-gov-east-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-gov-east-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- AWS GovCloud (US-West) (us-gov-west-1)

```
https://console.aws.amazon.com/cloudformation/home?region=us-gov-west-1#/stacks/
create/review?stackName=hpc-networking&templateURL=https://aws-hpc-recipes.s3.us-
east-1.amazonaws.com/main/recipes/net/hpc_large_scale/assets/main.yaml
```

- Nur Vorlage

```
https://aws-hpc-recipes.s3.us-east-1.amazonaws.com/main/recipes/net/hpc_large_scale/
assets/main.yaml
```

Um eine Amazon VPC für PCS zu erstellen

1. Öffnen Sie die Vorlage im CloudFormation Konsole.

   > **Note**
   >
   > Diese sind in der Vorlage vorausgefüllt, sodass Sie sie einfach als Standardwerte belassen können.

2. Unter Geben Sie einen Stack-Namen an, dann Name des Stapels, geben Sie ein `hpc-networking`.
3. Unter Parameter, geben Sie die folgenden Details ein:

   a. Unter VPC, dann CidrBlock, geben Sie ein `10.3.0.0/16`

   b. Unter Subnetze A:

      i. Dann CidrPublicSubnetA, geben Sie ein `10.3.0.0/20`

ii. Dann CidrPrivateSubnetA, geben Sie ein `10.3.128.0/20`

c. Unter Subnetze B:

i. Dann CidrPublicSubnetB, geben Sie ein `10.3.16.0/20`

ii. Dann CidrPrivateSubnetA, geben Sie ein `10.3.144.0/20`

d. Unter Subnetze C:

i. Für ProvisionSubnetsC, wählen `True`.

> **Note**
>
> Wenn Sie eine VPC in einer Region mit weniger als drei Availability Zones erstellen, wird diese Option ignoriert, wenn sie auf gesetzt ist `True`.

ii. Dann CidrPublicSubnetB, geben Sie ein `10.3.32.0/20`

iii. Dann CidrPrivateSubnetA, geben Sie ein `10.3.160.0/20`

4. Unter Fähigkeiten, kreuzen Sie das Kästchen für Ich erkenne an, dass AWS CloudFormation könnte IAM-Ressourcen erstellen.

Überwachen Sie den Status des CloudFormation stapeln. Wenn es erreicht `CREATE_COMPLETE`, die VPC-Ressourcen sind für Sie einsatzbereit.

> **Note**
>
> Um alle Ressourcen zu sehen, CloudFormation Die erstellte Vorlage öffnen Sie die CloudFormation Konsole. Wählen Sie das `hpc-networking`-Stack, und wählen Sie dann die Registerkarte Ressourcen.

Sicherheitsgruppen in AWS STK.

Sicherheitsgruppen in Amazon EC2 dienen als virtuelle Firewalls zur Steuerung des ein- und ausgehenden Datenverkehrs zu Instances. Verwenden Sie eine Startvorlage für eine AWS PCS-Compute-Knotengruppe, um Sicherheitsgruppen zu ihren Instances hinzuzufügen oder zu entfernen. Wenn Ihre Startvorlage keine Netzwerkschnittstellen enthält, verwenden Sie diese, `SecurityGroupIds` um eine Liste von Sicherheitsgruppen bereitzustellen. Wenn Ihre Startvorlage Netzwerkschnittstellen definiert, müssen Sie den `Groups` Parameter verwenden, um jeder

Netzwerkschnittstelle Sicherheitsgruppen zuzuweisen. Weitere Informationen über Startvorlagen finden Sie unter Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS.

> **Note**
>
> Änderungen an der Sicherheitsgruppenkonfiguration in der Startvorlage wirken sich nur auf neue Instances aus, die nach der Aktualisierung der Compute-Knotengruppe gestartet werden.

Anforderungen und Überlegungen zur Sicherheitsgruppe

AWS PCS erstellt ein kontenübergreifendes Elastic Network Interface (ENI) in dem Subnetz, das Sie bei der Erstellung eines Clusters angeben. Dies bietet dem HPC-Scheduler, der in einem von PCS verwalteten Konto ausgeführt wird AWS, einen Pfad für die Kommunikation mit EC2-Instances, die von PCS gestartet wurden. AWS Sie müssen eine Sicherheitsgruppe für diese ENI bereitstellen, die eine bidirektionale Kommunikation zwischen dem Scheduler-ENI und Ihren Cluster-EC2-Instances ermöglicht.

Eine einfache Möglichkeit, dies zu erreichen, besteht darin, eine permissive, selbstreferenzierende Sicherheitsgruppe zu erstellen, die den TCP/IP Datenverkehr auf allen Ports zwischen allen Mitgliedern der Gruppe zulässt. Sie können dies sowohl an den Cluster- als auch an die EC2-Instances der Knotengruppe anhängen.

Beispiel für eine permissive Sicherheitsgruppenkonfiguration

IPv4

Art der Regel	Protokolle	Ports	Quelle	Ziel
Eingehend	Alle	Alle	Selbst	
Ausgehend	Alle	Alle		0.0.0. 0/0
Ausgehend	Alle	Alle		Selbst

IPv6

Art der Regel	Protokolle	Ports	Quelle	Ziel
Eingehend	Alle	Alle	Selbst	
Ausgehend	Alle	Alle		::/0
Ausgehend	Alle	Alle		Selbst

Diese Regeln ermöglichen den freien Fluss des gesamten Datenverkehrs zwischen dem Slurm-Controller und den Knoten, lassen den gesamten ausgehenden Verkehr zu einem beliebigen Ziel zu und ermöglichen EFA-Verkehr.

Beispiel für eine restriktive Sicherheitsgruppenkonfiguration

Sie können auch die offenen Ports zwischen dem Cluster und seinen Rechenknoten einschränken. Für den Slurm-Scheduler muss die mit Ihrem Cluster verbundene Sicherheitsgruppe die folgenden Ports zulassen:

- 6817 — aktiviert eingehende Verbindungen zu EC2-Instances `slurmctld`
- 6818 — ermöglicht ausgehende Verbindungen von `slurmctld` und zur Ausführung auf EC2-Instances `slurmd`

Die mit Ihren Rechenknoten verbundene Sicherheitsgruppe muss die folgenden Ports zulassen:

- 6817 — ermöglicht ausgehende Verbindungen zu `slurmctld` EC2-Instances.
- 6818 — ermöglicht eingehende und ausgehende Verbindungen `slurmd` von `slurmctld` und zu Knotengruppen-Instances `slurmd`
- AWS PCS legt standardmäßig keinen festen `srun` Portbereich fest und `srun` verwendet daher die kurzlebigen Ports des Betriebssystems (unter Linux 32768—60999), sofern Sie nicht die SrunPortRangebenutzerdefinierte Slurm-Einstellung angeben (z. B.) und diesen Bereich in Ihren Sicherheitsgruppenregeln zulassen. `SrunPortRange=60001-63000` Siehe Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster.
- EFA-Verkehr zwischen Knotengruppen-Instances. Weitere Informationen finden Sie unter Vorbereiten einer EFA-enabled Sicherheitsgruppe im Benutzerhandbuch für Linux-Instances
- Jeder andere Datenverkehr zwischen den Knoten, der für Ihren Workload erforderlich ist

Mehrere Netzwerkschnittstellen in AWS PCS

Einige EC2-Instances haben mehrere Netzwerkkarten. Dadurch können sie eine höhere Netzwerkleistung bieten, einschließlich Bandbreitenkapazitäten von über 100 Gbit/s und verbesserter Paketverarbeitung. Weitere Informationen zu Instances mit mehreren Netzwerkkarten finden Sie unter Elastic Network Interfaces im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Konfigurieren Sie zusätzliche Netzwerkkarten für Instances in einer AWS PCS-Compute-Knotengruppe, indem Sie der zugehörigen EC2-Startvorlage Netzwerkschnittstellen hinzufügen. Im Folgenden finden Sie ein Beispiel für eine Startvorlage, die zwei Netzwerkkarten aktiviert, wie sie sich beispielsweise auf einer `hpc7a.96xlarge` Instance befinden. Beachten Sie folgende Details:

- Das Subnetz für jede Netzwerkschnittstelle muss das gleiche sein, das Sie bei der Konfiguration der AWS PCS-Compute-Knotengruppe ausgewählt haben, die die Startvorlage verwendet.
- Das primäre Netzwerkgerät, auf dem routinemäßige Netzwerkkommunikation wie SSH- und HTTPS-Verkehr stattfindet, wird durch die Einstellung von eingerichtet. `DeviceIndex` `0` Andere Netzwerkschnittstellen haben einen Wert `DeviceIndex` von. `1` Es kann nur eine primäre Netzwerkschnittstelle geben — alle anderen Schnittstellen sind sekundär.
- Alle Netzwerkschnittstellen müssen eindeutig sein. `NetworkCardIndex` Es wird empfohlen, sie sequenziell zu nummerieren, so wie sie in der Startvorlage definiert sind.
- Sicherheitsgruppen für jede Netzwerkschnittstelle werden mithilfe von `Groups` festgelegt. In diesem Beispiel wird der primären Netzwerkschnittstelle eine eingehende SSH-Sicherheitsgruppe (`sg-`*`SshSecurityGroupId`*) hinzugefügt, ebenso wie die Sicherheitsgruppe, die die Kommunikation innerhalb des Clusters ermöglicht (). `sg-`*`ClusterSecurityGroupId`* Schließlich wird sowohl der primären als auch der sekundären Schnittstelle eine Sicherheitsgruppe hinzugefügt, die ausgehende Verbindungen zum Internet (`sg-`*`InternetOutboundSecurityGroupId`*) ermöglicht.

```
{
    "NetworkInterfaces": [
        {
            "DeviceIndex": 0,
            "NetworkCardIndex": 0,
            "SubnetId": "subnet-SubnetId",
            "Groups": [
               "sg-SshSecurityGroupId",
               "sg-ClusterSecurityGroupId",
```

```
                "sg-InternetOutboundSecurityGroupId"
            ]
        },
        {
            "DeviceIndex": 1,
            "NetworkCardIndex": 1,
            "SubnetId": "subnet-SubnetId",
            "Groups": ["sg-InternetOutboundSecurityGroupId"]
        }
    ]
}
```

Platzierungsgruppen für EC2-Instances in AWS PCS

Sie können eine Platzierungsgruppe verwenden, um die Platzierung von EC2-Instances so zu beeinflussen, dass sie den Anforderungen der Arbeitslast entspricht, die auf ihnen ausgeführt wird.

Typen von Platzierungsgruppen

- Cluster — Fügt Instanzen nahe beieinander in einer Availability Zone zusammen, um die Kommunikation mit niedriger Latenz zu optimieren.
- Partition — Verteilt Instanzen auf logische Partitionen, um die Ausfallsicherheit zu maximieren.
- Verteilung — Erzwingt strikt, dass eine kleine Anzahl von Instances auf unterschiedlicher Hardware gestartet wird, was auch zur Erhöhung der Ausfallsicherheit beitragen kann.

Weitere Informationen finden Sie unter Platzierungsgruppen für Ihre Amazon EC2 EC2-Instances im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Wir empfehlen, eine Cluster-Platzierungsgruppe einzubeziehen, wenn Sie eine AWS PCS-Compute-Knotengruppe für die Verwendung des Elastic Fabric Adapter (EFA) konfigurieren.

Um eine Cluster-Platzierungsgruppe zu erstellen, die mit EFA funktioniert

1. Erstellen Sie eine Platzierungsgruppe mit dem Typ Cluster für die Compute-Knotengruppe.
   - Verwenden Sie den folgenden AWS CLI Befehl:

     ```
     aws ec2 create-placement-group --strategy cluster --group-name PLACEMENT-GROUP-NAME
     ```

   - Sie können auch eine CloudFormation Vorlage verwenden, um eine Platzierungsgruppe zu erstellen. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter

Arbeiten mit CloudFormation Vorlagen. Laden Sie die Vorlage von der folgenden URL herunter und laden Sie sie in die CloudFormation Konsole hoch.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-placement-group.yaml
```

2. Nehmen Sie die Platzierungsgruppe in die EC2-Startvorlage für die AWS PCS-Compute-Knotengruppe auf.

Verwenden des Elastic Fabric Adapter (EFA) mit AWS PCS

Elastic Fabric Adapter (EFA) ist eine leistungsstarke, fortschrittliche Netzwerkverbindung AWS , die Sie an Ihre EC2-Instance anschließen können, um High Performance Computing (HPC) und Machine-Learning-Anwendungen zu beschleunigen. Um Ihre Anwendungen, die auf einem AWS PCS-Cluster mit EFA ausgeführt werden, zu aktivieren, müssen Sie die Instances der AWS PCS-Compute-Knotengruppe so konfigurieren, dass EFA wie folgt verwendet wird.

> **Note**
>
> EFA auf einem AWS PCS-compatible AMI installieren — Auf dem in der AWS PCS-Compute-Knotengruppe verwendeten AMI muss der EFA-Treiber installiert und geladen sein. Informationen zum Erstellen eines benutzerdefinierten AMI mit installierter EFA-Software finden Sie unterBenutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Inhalt

- Identifizieren Sie EFA-enabled EC2-Instances
- Erstellen Sie eine Sicherheitsgruppe zur Unterstützung der EFA-Kommunikation
- (Optional) Erstellen Sie eine Platzierungsgruppe
- Erstellen oder aktualisieren Sie eine EC2-Startvorlage
- Erstellen oder aktualisieren Sie Compute-Knotengruppen für EFA
- (Optional) Testen Sie EFA
- (Optional) Verwenden Sie eine CloudFormation Vorlage, um eine EFA-enabled Startvorlage zu erstellen

Identifizieren Sie EFA-enabled EC2-Instances

Um EFA verwenden zu können, müssen alle Instance-Typen, die für eine AWS PCS-Compute-Gruppe zulässig sind, EFA unterstützen und dieselbe Anzahl von vCPUs (und GPUs, falls zutreffend) haben. Eine Liste der EFA-enabled Instances finden Sie unter Elastic Fabric Adapter für HPC- und ML-Workloads auf Amazon EC2 im Amazon Elastic Compute Cloud-Benutzerhandbuch. Sie können den auch verwenden AWS CLI , um eine Liste der Instance-Typen anzuzeigen, die EFA unterstützen. *region-code*Ersetzen Sie es durch den AWS-Region Ort, an dem Sie AWS PCS verwenden, z. B. `us-east-1`

```
aws ec2 describe-instance-types \
   --region region-code \
   --filters Name=network-info.efa-supported,Values=true \
   --query "InstanceTypes[*].[InstanceType]" \
   --output text | sort
```

> **Note**
>
> Ermitteln Sie, wie viele Netzwerkschnittstellen verfügbar sind — Einige EC2-Instances verfügen über mehrere Netzwerkkarten. Dadurch können sie über mehrere EFAs verfügen. Weitere Informationen finden Sie unter Mehrere Netzwerkschnittstellen in AWS PCS.

Erstellen Sie eine Sicherheitsgruppe zur Unterstützung der EFA-Kommunikation

AWS CLI

Sie können den folgenden AWS CLI Befehl verwenden, um eine Sicherheitsgruppe zu erstellen, die EFA unterstützt. Der Befehl gibt eine Sicherheitsgruppen-ID aus. Nehmen Sie die folgenden Ersetzungen vor:

- *region-code*— Geben Sie an AWS-Region , wo Sie AWS PCS verwenden, z. B. `us-east-1`
- *vpc-id*— Geben Sie die ID der VPC an, die Sie für AWS PCS verwenden.
- *efa-group-name*— Geben Sie den von Ihnen gewählten Namen für die Sicherheitsgruppe ein.

```
aws ec2 create-security-group \
    --group-name efa-group-name \
    --description "Security group to enable EFA traffic" \
    --vpc-id vpc-id \
    --region region-code
```

Verwenden Sie die folgenden Befehle, um Sicherheitsgruppenregeln für eingehenden und ausgehenden Datenverkehr anzuhängen. Nehmen Sie den folgenden Ersatz vor:

- *efa-secgroup-id*— Geben Sie die ID der EFA-Sicherheitsgruppe an, die Sie gerade erstellt haben.

```
aws ec2 authorize-security-group-ingress \
    --group-id efa-secgroup-id \
    --protocol -1 \
    --source-group efa-secgroup-id

aws ec2 authorize-security-group-egress \
    --group-id efa-secgroup-id \
    --protocol -1 \
    --source-group efa-secgroup-id
```

CloudFormation template

Sie können eine CloudFormation Vorlage verwenden, um eine Sicherheitsgruppe zu erstellen, die EFA unterstützt. Laden Sie die Vorlage von der folgenden URL herunter und laden Sie sie dann in die AWS CloudFormation Konsole hoch.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/efa-sg.yaml
```

Geben Sie bei geöffneter Vorlage in der AWS CloudFormation Konsole die folgenden Optionen ein.

- Unter Geben Sie einen Stacknamen an
  - Geben Sie unter Stackname einen Namen ein, z. `efa-sg-stack` B.
- Unter Parameter
  - Geben Sie SecurityGroupNameunter einen Namen ein, z. `efa-sg` B.

- Wählen Sie unter VPC die VPC aus, in der Sie PCS verwenden AWS möchten.

Beenden Sie die Erstellung des CloudFormation Stacks und überwachen Sie seinen Status. Wenn es erreicht ist, ist `CREATE_COMPLETE` die EFA-Sicherheitsgruppe einsatzbereit.

(Optional) Erstellen Sie eine Platzierungsgruppe

Wir empfehlen, alle Instances, die EFA verwenden, in einer Cluster-Placement-Gruppe zu starten, um die physische Entfernung zwischen ihnen zu minimieren. Erstellen Sie eine Platzierungsgruppe für jede Rechenknotengruppe, in der Sie EFA verwenden möchten. Informationen Platzierungsgruppen für EC2-Instances in AWS PCS zum Erstellen einer Platzierungsgruppe für Ihre Compute-Knotengruppe finden Sie unter.

Erstellen oder aktualisieren Sie eine EC2-Startvorlage

EFA-Netzwerkschnittstellen werden in der EC2-Startvorlage für eine AWS PCS-Compute-Knotengruppe eingerichtet. Wenn mehrere Netzwerkkarten vorhanden sind, können mehrere EFAs konfiguriert werden. Die EFA-Sicherheitsgruppe und die optionale Platzierungsgruppe sind ebenfalls in der Startvorlage enthalten.

Hier ist ein Beispiel für eine Startvorlage für Instances mit zwei Netzwerkkarten, z. B. hpc7a.96xlarge. Die Instances werden in einer Cluster-Platzierungsgruppe gestartet. `subnet-`*`SubnetID1`* `pg-`*`PlacementGroupId1`*

Sicherheitsgruppen müssen jeder EFA-Schnittstelle speziell hinzugefügt werden. Jede EFA benötigt die Sicherheitsgruppe, die den EFA-Verkehr aktiviert (). `sg-`*`EfaSecGroupId`* Andere Sicherheitsgruppen, insbesondere solche, die regulären Datenverkehr wie SSH oder HTTPS verarbeiten, müssen nur an die primäre Netzwerkschnittstelle (gekennzeichnet durch ein `DeviceIndex` of) angehängt werden. `0` Startvorlagen, in denen Netzwerkschnittstellen definiert sind, unterstützen die Einstellung von Sicherheitsgruppen mithilfe des `SecurityGroupIds` Parameters nicht. Sie müssen `Groups` in jeder Netzwerkschnittstelle, die Sie konfigurieren, einen Wert für festlegen.

```
{
    "Placement": {
        "GroupId": "pg-PlacementGroupId1"
    },
    "NetworkInterfaces": [
```

```
        {
            "DeviceIndex": 0,
            "InterfaceType": "efa",
            "NetworkCardIndex": 0,
            "SubnetId": "subnet-SubnetId1",
            "Groups": [
                "sg-SecurityGroupId1",
                "sg-EfaSecGroupId"
            ]
        },
        {
            "DeviceIndex": 1,
            "InterfaceType": "efa",
            "NetworkCardIndex": 1,
            "SubnetId": "subnet-SubnetId1"
            "Groups": ["sg-EfaSecGroupId"]
        }
    ]
}
```

Erstellen oder aktualisieren Sie Compute-Knotengruppen für EFA

Ihre AWS PCS-Compute-Knotengruppen müssen Instances mit derselben Anzahl von vCPUs, Prozessorarchitektur und EFA-Unterstützung enthalten. Konfigurieren Sie die Compute-Knotengruppe so, dass sie das AMI mit der darauf installierten EFA-Software verwendet und die Startvorlage verwendet, mit der EFA-enabled Netzwerkschnittstellen konfiguriert werden.

(Optional) Testen Sie EFA

Sie können die EFA-enabled Kommunikation zwischen zwei Knoten in einer Rechenknotengruppe demonstrieren, indem Sie `fi_pingpong` das Programm ausführen, das in der EFA-Softwareinstallation enthalten ist. Wenn dieser Test erfolgreich ist, ist EFA wahrscheinlich richtig konfiguriert.

Zu Beginn benötigen Sie zwei laufende Instances in der Compute-Knotengruppe. Wenn Ihre Compute-Knotengruppe statische Kapazität verwendet, sollten bereits Instanzen verfügbar sein. Für eine Rechenknotengruppe, die dynamische Kapazität verwendet, können Sie mit dem `salloc` Befehl zwei Knoten starten. Hier ist ein Beispiel aus einem Cluster mit einer dynamischen Knotengruppe namens, die einer Warteschlange mit dem Namen `hpc7g` zugeordnet ist`all`.

```
% salloc --nodes 2 -p all
```

```
salloc: Granted job allocation 6
salloc: Waiting for resource configuration
... a few minutes pass ...
salloc: Nodes hpc7g-[1-2] are ready for job
```

Ermitteln Sie die IP-Adresse für die beiden zugewiesenen Knoten mithilfe von`scontrol`. Im folgenden Beispiel sind die Adressen `10.3.140.69` für `hpc7g-1` und `10.3.132.211` für`hpc7g-2`.

```
% scontrol show nodes hpc7g-[1-2]
NodeName=hpc7g-1 Arch=aarch64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=hpc7g
   ActiveFeatures=hpc7g
   Gres=(null)
   NodeAddr=10.3.140.69 NodeHostName=ip-10-3-140-69 Version=25.11.2
   OS=Linux 6.12.80-106.156.amzn2023.aarch64 #1 SMP Fri May 1 14:08:14 UTC 2026
   RealMemory=124518 AllocMem=0 FreeMem=110763 Sockets=64 Boards=1
   State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=efa
   BootTime=2026-05-02T19:00:09 SlurmdStartTime=2026-05-08T19:33:25
   LastBusyTime=2026-05-08T19:33:25 ResumeAfterTime=None
   CfgTRES=cpu=64,mem=124518M,billing=64
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
   Reason=Maintain Minimum Number Of Instances [root@2026-05-02T18:59:00]
   InstanceId=i-04927897a9ce3c143 InstanceType=hpc7g.16xlarge

NodeName=hpc7g-2 Arch=aarch64 CoresPerSocket=1
   CPUAlloc=0 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=hpc7g
   ActiveFeatures=hpc7g
   Gres=(null)
   NodeAddr=10.3.132.211 NodeHostName=ip-10-3-132-211 Version=25.11.2
   OS=Linux 6.12.80-106.156.amzn2023.aarch64 #1 SMP Fri May 1 14:08:14 UTC 2026
   RealMemory=124518 AllocMem=0 FreeMem=110759 Sockets=64 Boards=1
   State=IDLE+CLOUD ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=efa
   BootTime=2026-05-02T19:00:09 SlurmdStartTime=2026-05-08T19:33:25
   LastBusyTime=2026-05-08T19:33:25 ResumeAfterTime=None
   CfgTRES=cpu=64,mem=124518M,billing=64
   AllocTRES=
```

```
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/a ExtSensorsWatts=0 ExtSensorsTemp=n/a
   Reason=Maintain Minimum Number Of Instances [root@2026-05-02T18:59:00]
   InstanceId=i-0a2c82623cb1393a7 InstanceType=hpc7g.16xlarge
```

Stellen Sie mithilfe von SSH (oder SSMhpc7g-1) eine Connect zu einem der Knoten her (in diesem Beispielfall). Beachten Sie, dass es sich um eine interne IP-Adresse handelt. Wenn Sie SSH verwenden, müssen Sie daher möglicherweise eine Verbindung von einem Ihrer Anmeldeknoten aus herstellen. Beachten Sie auch, dass die Instanz mithilfe der Startvorlage für Compute-Knotengruppen mit einem SSH-Schlüssel konfiguriert werden muss.

```
% ssh ec2-user@10.3.140.69
```

Starten Sie jetzt `fi_pingpong` im Servermodus.

```
/opt/amazon/efa/bin/fi_pingpong -p efa
```

Connect zur zweiten Instanz her (`hpc7g-2`).

```
% ssh ec2-user@10.3.132.211
```

Führen Sie `fi_pingpong` im Client-Modus aus und stellen Sie eine Verbindung zum Server her`hpc7g-1`. Sie sollten eine Ausgabe sehen, die dem Beispiel unten ähnelt.

```
% /opt/amazon/efa/bin/fi_pingpong -p efa 10.3.140.69

bytes   #sent   #ack     total       time     MB/sec    usec/xfer   Mxfers/sec
64      10      =10      1.2k        0.00s      3.08      20.75       0.05
256     10      =10      5k          0.00s     21.24      12.05       0.08
1k      10      =10      20k         0.00s     82.91      12.35       0.08
4k      10      =10      80k         0.00s    311.48      13.15       0.08
[error] util/pingpong.c:1876: fi_close (-22) fid 0
```

(Optional) Verwenden Sie eine CloudFormation Vorlage, um eine EFA-enabled Startvorlage zu erstellen

Da die Einrichtung von EFA mit mehreren Abhängigkeiten verbunden ist, wurde eine CloudFormation Vorlage bereitgestellt, mit der Sie eine Rechenknotengruppe konfigurieren können. Sie unterstützt

Instanzen mit bis zu vier Netzwerkkarten. Weitere Informationen zu Instances mit mehreren Netzwerkkarten finden Sie unter Elastic Network Interfaces im Amazon Elastic Compute Cloud-Benutzerhandbuch.

Laden Sie die CloudFormation Vorlage von der folgenden URL herunter und laden Sie sie dann auf die CloudFormation Konsole hoch, AWS-Region in der Sie AWS PCS verwenden.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/enable_efa/assets/pcs-lt-efa.yaml
```

Geben Sie bei geöffneter Vorlage in der CloudFormation Konsole die folgenden Werte ein. Beachten Sie, dass die Vorlage einige Standardparameterwerte bereitstellt. Sie können sie als Standardwerte beibehalten.

- Unter Geben Sie einen Stacknamen an
  - Geben Sie unter Stackname einen beschreibenden Namen ein. Wir empfehlen, den Namen zu verwenden, den Sie für Ihre AWS PCS-Compute-Knotengruppe wählen werden, z. B. *NODEGROUPNAME*`-efa-lt`
- Unter Parameter
  - Wählen Sie unter NumberOfNetworkCardsdie Anzahl der Netzwerkkarten in den Instanzen aus, die zu Ihrer Knotengruppe gehören sollen.
  - Wählen Sie unter die VPC aus VpcId, auf der Ihr AWS PCS-Cluster bereitgestellt wird.
  - Wählen Sie NodeGroupSubnetIdunter das Subnetz in Ihrer Cluster-VPC aus, in dem EFA-enabled Instances gestartet werden.
  - Lassen Sie das Feld unter leer PlacementGroupName, um eine neue Cluster-Platzierungsgruppe für die Knotengruppe zu erstellen. Wenn Sie bereits über eine Platzierungsgruppe verfügen, die Sie verwenden möchten, geben Sie hier ihren Namen ein.
  - Wählen Sie unter die Sicherheitsgruppe aus ClusterSecurityGroupId, die Sie verwenden, um den Zugriff auf andere Instances im Cluster und auf die AWS PCS-API zu gewähren. Viele Kunden wählen die Standardsicherheitsgruppe aus ihrer Cluster-VPC.
  - Geben Sie unter die ID für eine Sicherheitsgruppe ein SshSecurityGroupId, die Sie verwenden, um eingehenden SSH-Zugriff auf Knoten in Ihrem Cluster zu ermöglichen.
  - Wählen Sie für SshKeyNamedas SSH-Schlüsselpaar für den Zugriff auf Knoten in Ihrem Cluster aus.

  - Geben Sie für LaunchTemplateNameeinen beschreibenden Namen für die Startvorlage ein, z. B. *NODEGROUPNAME*`-efa-lt` Der Name muss für den Ort, AWS-Konto an AWS-Region dem Sie AWS PCS verwenden werden, einzigartig sein.
- Unter Funktionen
  - Markieren Sie das Kästchen Ich bestätige, dass dadurch IAM-Ressourcen erstellt werden AWS CloudFormation könnten.

Überwachen Sie den Status des CloudFormation Stacks. Wenn `CREATE_COMPLETE` die Startvorlage erreicht ist, kann sie verwendet werden. Verwenden Sie es mit einer AWS PCS-Compute-Knotengruppe, wie oben unter beschriebenErstellen oder aktualisieren Sie Compute-Knotengruppen für EFA.

Verwenden von Netzwerkdateisystemen mit AWS PCS

Sie können Netzwerkdateisysteme an Knoten anhängen, die in einer AWS PCS-Rechenknotengruppe (AWS Parallel Computing Service) gestartet wurden, um einen dauerhaften Speicherort bereitzustellen, an dem Daten und Dateien geschrieben und abgerufen werden können. Sie können Dateisysteme verwenden, die von AWS Diensten wie Amazon Elastic File System (Amazon EFS), Amazon FSx for Lustre, Amazon for NetApp ONTAP, Amazon FSx FSx forOpenZFS und Amazon File Cache bereitgestellt werden. Sie können auch selbstverwaltete Dateisysteme wie NFS-Server verwenden.

In diesem Thema werden Überlegungen und Beispiele für die Verwendung von Netzwerkdateisystemen mit AWS PCS behandelt.

Überlegungen zur Verwendung von Netzwerkdateisystemen

Die Implementierungsdetails für verschiedene Dateisysteme sind unterschiedlich, es gibt jedoch einige allgemeine Überlegungen.

- Die entsprechende Dateisystemsoftware muss auf der Instanz installiert sein. Um beispielsweise Amazon FSx for Lustre zu verwenden, sollte das entsprechende Lustre Paket vorhanden sein. Dies kann erreicht werden, indem es in das Compute-Knotengruppen-AMI aufgenommen wird oder indem ein Skript verwendet wird, das beim Instance-Start ausgeführt wird.
- Es muss eine Netzwerkroute zwischen dem gemeinsam genutzten Netzwerkdateisystem und den Compute-Knotengruppen-Instances bestehen.
- Die Sicherheitsgruppenregeln sowohl für das gemeinsam genutzte Netzwerk-Dateisystem als auch für die Compute-Knotengruppen-Instanzen müssen Verbindungen zu den entsprechenden Ports zulassen.
- Sie müssen für alle Ressourcen, die auf die Dateisysteme zugreifen, einen konsistenten POSIX Benutzer- und Gruppennamespace aufrechterhalten. Andernfalls kann es bei Aufträgen und interaktiven Prozessen, die auf Ihrem PCS-Cluster ausgeführt werden, zu Berechtigungsfehlern kommen.
- Das Einhängen von Dateisystemen erfolgt mithilfe von EC2 Startvorlagen. Fehler oder Zeitüberschreitungen beim Mounten eines Netzwerkdateisystems können dazu führen, dass Instanzen nicht mehr für die Ausführung von Jobs verfügbar sind. Dies wiederum kann zu unerwarteten Kosten führen. Weitere Informationen zum Debuggen von Startvorlagen finden Sie unterVerwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS.

Beispiele für Netzwerk-Mounts

Sie können Dateisysteme mit Amazon EFS, Amazon FSx for Lustre, Amazon for NetApp ONTAP, Amazon FSx FSx for OpenZFS und Amazon File Cache erstellen. Erweitern Sie den entsprechenden Abschnitt unten, um ein Beispiel für jeden Netzwerk-Mount zu sehen.

Amazon EFS

Einrichtung des Dateisystems

Erstellen Sie ein Amazon EFS-Dateisystem. Stellen Sie sicher, dass es in jeder Availability Zone, in der Sie PCS-Compute-Knotengruppen-Instances starten, ein Mount-Ziel gibt. Stellen Sie außerdem sicher, dass jedes Mount-Ziel einer Sicherheitsgruppe zugeordnet ist, die eingehenden und ausgehenden Zugriff von den PCS-Compute-Knotengruppen-Instances aus ermöglicht. Weitere Informationen finden Sie unter Bereitstellen von Zielen und Sicherheitsgruppen im Amazon Elastic File System-Benutzerhandbuch.

Startvorlage

Fügen Sie die Sicherheitsgruppe (n) aus Ihrem Dateisystem-Setup zur Startvorlage hinzu, die Sie für die Compute-Knotengruppe verwenden werden.

Fügen Sie Benutzerdaten hinzu, die `cloud-config` einen Mechanismus zum Mounten des Amazon EFS-Dateisystems verwenden. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *`mount-point-directory`*— Der Pfad auf jeder Instance, auf der Sie Amazon EFS mounten werden
- *`filesystem-id`*— Die Dateisystem-ID für das EFS-Dateisystem

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

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

packages:
  - amazon-efs-utils
```

```
runcmd:
  - mkdir -p /mount-point-directory
  - echo "filesystem-id:/ /mount-point-directory efs tls,_netdev" >> /etc/fstab
  - mount -a -t efs defaults

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

Amazon FSx für Lustre

Einrichtung des Dateisystems

Erstellen Sie ein FSx for Lustre-Dateisystem in der VPC, in dem Sie PCS verwenden AWS werden. Um Übertragungen zwischen Zonen zu minimieren, sollten Sie die Implementierung in einem Subnetz in derselben Availability Zone durchführen, in der Sie die meisten Ihrer PCS-Compute-Knotengruppen-Instances starten werden. Stellen Sie sicher, dass das Dateisystem einer Sicherheitsgruppe zugeordnet ist, die eingehenden und ausgehenden Zugriff von den PCS-Compute-Knotengruppen-Instances aus ermöglicht. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Dateisystem-Zugriffskontrolle mit Amazon VPC im Amazon FSx for Lustre-Benutzerhandbuch.

Startvorlage

Fügen Sie Benutzerdaten hinzu, die `cloud-config` zum Mounten des FSx for Lustre-Dateisystems verwendet werden. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *`mount-point-directory`*— Der Pfad auf einer Instanz, die Sie FSx für Lustre mounten möchten
- *`filesystem-id`*— Die Dateisystem-ID für das FSx for Lustre-Dateisystem
- *`mount-name`*— Der Mount-Name für das FSx for Lustre-Dateisystem
- *`region-code`*— Der AWS-Region Ort, an dem das FSx for Lustre-Dateisystem bereitgestellt wird (muss mit Ihrem AWS PCS-System identisch sein)
- (Optional) *`latest`* — Jede Version von, die von FSx for Lustre Lustre unterstützt wird

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

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

runcmd:
- amazon-linux-extras install -y lustre=latest
```

```
- mkdir -p /mount-point-directory
- mount -t lustre filesystem-id.fsx.region-code.amazonaws.com@tcp:/mount-name /mount-
point-directory

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

Amazon FSx für NetApp ONTAP

Einrichtung des Dateisystems

Erstellen Sie ein Amazon FSx for NetApp ONTAP-Dateisystem in der VPC, in der Sie PCS verwenden AWS werden. Um Übertragungen zwischen Zonen zu minimieren, sollten Sie die Bereitstellung in einem Subnetz in derselben Availability Zone durchführen, in der Sie die meisten Ihrer AWS PCS-Compute-Knotengruppen-Instances starten werden. Stellen Sie sicher, dass das Dateisystem einer Sicherheitsgruppe zugeordnet ist, die eingehenden und ausgehenden Zugriff von den Instances der AWS PCS-Compute-Knotengruppe aus ermöglicht. Weitere Informationen zu Sicherheitsgruppen finden Sie unter File System Access Control with Amazon VPC im FSx for ONTAP User Guide.

Startvorlage

Fügen Sie Benutzerdaten hinzu, die `cloud-config` zum Mounten des Root-Volumes FSx für ein ONTAP-Dateisystem verwendet werden. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *`mount-point-directory`*— Der Pfad auf einer Instance, auf der Sie Ihr FSx for ONTAP-Volume mounten möchten
- *`svm-id`*— Die SVM-ID für das Dateisystem FSx für ONTAP
- *`filesystem-id`*— Die Dateisystem-ID FSx für das ONTAP-Dateisystem
- *`region-code`*— Der AWS-Region Ort, an dem das Dateisystem FSx für ONTAP bereitgestellt wird (muss mit Ihrem AWS PCS-System identisch sein)
- *`volume-name`*— Der Name des FSx Volumes für ONTAP

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

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

```
runcmd:
- mkdir -p /mount-point-directory
- mount -t nfs svm-id.filesystem-id.fsx.region-code.amazonaws.com:/volume-name /mount-
point-directory

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

Amazon FSx für OpenZFS

Einrichtung des Dateisystems

Erstellen Sie ein Dateisystem FSx für OpenZFS in der VPC, in dem Sie PCS verwenden werden. AWS Um Übertragungen zwischen Zonen zu minimieren, sollten Sie die Implementierung in einem Subnetz in derselben Availability Zone durchführen, in der Sie die meisten Ihrer AWS PCS-Compute-Knotengruppen-Instances starten werden. Stellen Sie sicher, dass das Dateisystem einer Sicherheitsgruppe zugeordnet ist, die eingehenden und ausgehenden Zugriff von den Instances der AWS PCS-Compute-Knotengruppe aus ermöglicht. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Verwaltung des Dateisystemzugriffs mit Amazon VPC im FSx OpenZFS-Benutzerhandbuch.

Startvorlage

Fügen Sie Benutzerdaten hinzu, die `cloud-config` zum Mounten des Root-Volumes FSx für ein OpenZFS-Dateisystem verwendet werden. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *`mount-point-directory`*— Der Pfad auf einer Instanz, auf der Sie Ihr FSx für OpenZFS Share mounten möchten
- *`filesystem-id`*— Die Dateisystem-ID für das Dateisystem FSx für OpenZFS
- *`region-code`*— Der AWS-Region Ort, an dem das Dateisystem FSx für OpenZFS bereitgestellt wird (muss mit Ihrem PCS-System identisch sein) AWS

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

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

```
runcmd:
- mkdir -p /mount-point-directory
- mount -t nfs -o noatime,nfsvers=4.2,sync,rsize=1048576,wsize=1048576 filesystem-
id.fsx.region-code.amazonaws.com:/fsx/ /mount-point-directory

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

Amazon-Datei-Cache

Einrichtung des Dateisystems

Erstellen Sie einen Amazon File Cache in der VPC, in der Sie AWS PCS verwenden werden. Um Übertragungen zwischen Zonen zu minimieren, wählen Sie ein Subnetz in derselben Availability Zone aus, in der Sie die meisten Ihrer PCS-Compute-Knotengruppen-Instances starten werden. Stellen Sie sicher, dass der Datei-Cache einer Sicherheitsgruppe zugeordnet ist, die eingehenden und ausgehenden Datenverkehr auf Port 988 zwischen Ihren PCS-Instances und dem File Cache zulässt. Weitere Informationen zu Sicherheitsgruppen finden Sie unter Cache-Zugriffskontrolle mit Amazon VPC im Amazon File Cache-Benutzerhandbuch.

Startvorlage

Fügen Sie die Sicherheitsgruppe (n) aus Ihrem Dateisystem-Setup zur Startvorlage hinzu, die Sie für die Compute-Knotengruppe verwenden werden.

Schließen Sie Benutzerdaten ein, die `cloud-config` zum Mounten des Amazon File Cache verwendet werden. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *`mount-point-directory`*— Der Pfad auf einer Instanz, die Sie FSx für Lustre mounten möchten
- *`cache-dns-name`*— Der DNS-Name (Domain Name System) für den Dateicache
- *`mount-name`*— Der Mount-Name für den Datei-Cache

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

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

runcmd:
- amazon-linux-extras install -y lustre=2.12
- mkdir -p /mount-point-directory
```

```
- mount -t lustre -o relatime,flock cache-dns-name@tcp:/mount-name /mount-point-
directory

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

Amazon Machine Images (AMIs) für AWS STK.

AWS PCS arbeitet mit AMIs, die Sie bereitstellen, und bietet so eine große Flexibilität bei der Software und Konfiguration der Knoten in Ihrem Cluster. Für Produktions AI/ML - und HPC-Workloads können Sie das AWS-maintained PCS-ready DLAMI verwenden. Wenn Sie AWS PCS evaluieren, können Sie ein Beispiel-AMI verwenden, das von bereitgestellt wird AWS. Sie können auch Ihre eigenen benutzerdefinierten AMIs erstellen, um die volle Kontrolle über die Software und Konfiguration auf Ihren Knoten zu haben.

Themen

- Verwenden von PCS-ready DLAMI mit AWS 5 STCK
- Verwenden von Amazon Machine Images (AMIs) als Beispiel mit AWS STCK.
- Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS
- Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK
- Versionshinweise für AWS PCS-Beispiel-AMIs

Verwenden von PCS-ready DLAMI mit AWS 5 STCK

AWS PCS-ready DLAMI Base GPU AMI (Ubuntu 24.04) ist ein AWS-maintained Amazon Machine Image zum Ausführen AI/ML und HPC-Workloads auf PCs. AWS Es bietet eine produktionsbereite Grundlage, sodass Sie Cluster in wenigen Minuten bereitstellen können, anstatt benutzerdefinierte AMIs zu erstellen und zu validieren.

Was ist enthalten?

PCS-ready DLAMI basiert auf dem Deep Learning Base GPU AMI (Ubuntu 24.04) und fügt die folgenden PCS-Komponenten hinzu: AWS

- PCS Agent — Der PCS-Cluster-Management-Agent AWS
- Slurm für AWS PCS — Mehrere unterstützte Slurm-Versionen sind vorinstalliert. Basierend auf der Konfiguration Ihres Clusters wird beim Start der Instance automatisch die richtige Version aktiviert.
- EFS-Dienstprogramme — Zum Mounten von Amazon EFS-Dateisystemen

Das Quell-DLAMI stellt das Betriebssystem (Ubuntu 24.04), die NVIDIA-GPU-Treiber, das CUDA-Toolkit, die EFA-Treiber, den Lustre-Client und andere grundlegende Infrastrukturen bereit.

Einzelheiten zu diesen Komponenten finden Sie in den Versionshinweisen zum Deep Learning AMI. https://docs.aws.amazon.com/dlami/latest/devguide/appendix-ami-release-notes.html

PCS-ready DLAMI ist sowohl für x86_64- als auch für arm64-Architekturen verfügbar.

> **Note**
>
> PCS-ready DLAMI beinhaltet keine Anwendungssoftware wie AI/ML Frameworks (,, JAX) PyTorch TensorFlow, Compiler oder mathematische Bibliotheken. Sie können Ihre Anwendungsebene auf gemeinsam genutzten Dateisystemen hinzufügen oder indem Sie ein benutzerdefiniertes AMI auf DLAMI aufbauen. PCS-ready

Das Beschreibungsfeld jedes AMI fasst seinen Inhalt zusammen, einschließlich des Quell-DLAMI, auf dem es basiert, der PCS-Agent-Version, der unterstützten Slurm-Versionen und der Version der EFS-Dienstprogramme. Sie können dieses Feld in der Amazon EC2-Konsole oder mithilfe der API anzeigen. `describe-images` Das Folgende ist ein Beispiel für einen Wert im Feld Beschreibung:

```
PCS-Ready DLAMI based on Deep Learning Base OSS Nvidia Driver GPU AMI (Ubuntu 24.04)
 20260522. PCS Agent: 1.4.0-1. Slurm: 24.11.7-1, 25.05.7-1, 25.11.2-1. EFS Utils: 2.4.2
```

Finden Sie den aktuellen PCS-ready DLAMI

AWS-Managementkonsole

Um PCS-ready DLAMI in der Konsole zu finden

1. Öffnen Sie die AWS PCS-Konsole und navigieren Sie, um eine Compute-Knotengruppe zu erstellen oder zu bearbeiten.
2. Wählen Sie im Abschnitt AMI-Auswahl PCS-ready AMIs aus.
3. Es erscheint ein Drop-down-Menü mit den verfügbaren PCS-ready DLAMIs, gefiltert nach der Architektur Ihres ausgewählten Instance-Typs.
4. Wählen Sie AWS PCS-ready DLAMI Base AMI (Ubuntu 24.04). Im Drop-down-Menü werden die AMI-ID und der vollständige AMI-Name unten als Referenz angezeigt.

AWS CLI

Sie können die neueste PCS-ready DLAMI AMI ID mithilfe des Amazon EC2 Systems Manager Parameter Store abrufen. Ersetzen Sie durch Ihre*region-code*. AWS-Region

- x86_64

```
aws ssm get-parameter --region region-code \
  --name /aws/service/pcs/ami/dlami-base-ubuntu2404/x86_64/latest/ami-id \
  --query "Parameter.Value" --output text
```

- arm64

```
aws ssm get-parameter --region region-code \
  --name /aws/service/pcs/ami/dlami-base-ubuntu2404/arm64/latest/ami-id \
  --query "Parameter.Value" --output text
```

Alternativ können Sie nach PCS-ready DLAMI anhand des Namensmusters suchen:

- x86_64

```
aws ec2 describe-images --region region-code --owners amazon \
  --filters 'Name=name,Values=aws-pcs-ready-dlami-base-ubuntu2404-x86_64-*' \
            'Name=state,Values=available' \
  --query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

- arm64

```
aws ec2 describe-images --region region-code --owners amazon \
  --filters 'Name=name,Values=aws-pcs-ready-dlami-base-ubuntu2404-arm64-*' \
            'Name=state,Values=available' \
  --query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId]' --output text
```

Verwenden Sie die AMI-ID, wenn Sie eine Rechenknotengruppe erstellen oder aktualisieren.

Erhalten Sie Benachrichtigungen für neue Versionen

Sie können Benachrichtigungen erhalten, wenn ein neuer PCS-ready DLAMI veröffentlicht wird. Benachrichtigungen werden mit Amazon SNS unter dem folgenden Thema veröffentlicht:

```
arn:aws:sns:us-west-2:265886551188:pcs-ready-dlami-release-notifications
```

Nachrichten zu diesem Thema werden veröffentlicht, wenn ein neuer PCS-ready DLAMI veröffentlicht wird.

Um DLAMI-Veröffentlichungsbenachrichtigungen zu PCS-ready abonnieren

1. Öffnen Sie die Amazon-SNS-Konsole.
2. Ändern Sie in der Navigationsleiste bei Bedarf AWS-Region die Option in US West (Oregon). Sie müssen die Region auswählen, in der die SNS-Benachrichtigung, die Sie abonnieren, erstellt wurde.
3. Wählen Sie im Navigationsbereich Abonnements und dann Abonnement erstellen aus.
4. Führen Sie im Dialogfeld Create subscription Folgendes aus:
    a. Geben Sie für Themen-ARN Folgendes ein: `arn:aws:sns:us-west-2:265886551188:pcs-ready-dlami-release-notifications`
    b. Wählen Sie unter Protocol (Protokoll) die Option Email (E-Mail) aus.
    c. Geben Sie für Endpoint die E-Mail-Adresse ein, an die Sie Benachrichtigungen erhalten möchten.
    d. Wählen Sie Create subscription (Abonnement erstellen) aus.
5. Sie erhalten eine Bestätigungs-E-Mail mit dem Betreff AWS-Benachrichtigung — Abonnementbestätigung. Öffnen Sie die E-Mail und wählen Sie Confirm subscription aus, um Ihr Abonnement abzuschließen.

Verwendung mit Infrastructure as Code

Der SSM-Parameterpfad bietet eine stabile Referenz, die immer in die neueste AMI-ID aufgelöst wird. Sie können dies in CloudFormation Vorlagen verwenden, um neue Versionen bei erneuter Bereitstellung automatisch zu übernehmen:

```
AmiId: '{{resolve:ssm:/aws/service/pcs/ami/dlami-base-ubuntu2404/x86_64/latest/ami-id}}'
```

Auf eine neue Version aktualisieren

AWS veröffentlicht aktualisierte PCS-ready DLAMI-Versionen, wenn das Deep Learning Base-GPU-Quell-AMI aktualisiert wird oder wenn PCS-Komponenten (PCS Agent oder Slurm for PCS) aktualisiert werden. Um Ihren Cluster zu aktualisieren, rufen Sie mithilfe des oben beschriebenen SSM-Parameters oder der Namenssuche die neueste AMI-ID ab und aktualisieren Sie dann jede Rechenknotengruppe, sodass sie auf die neue AMI-ID verweist.

Verwenden von Amazon Machine Images (AMIs) als Beispiel mit AWS STCK.

AWS stellt Beispiel-AMIs bereit, die Sie als Ausgangspunkt für die Arbeit mit AWS PCS verwenden können.

> **Important**
>
> Beispiel-AMIs dienen zu Demonstrationszwecken und werden nicht für Produktionsworkloads empfohlen.

> **Important**
>
> Compute-Knotengruppen, die mit AWS PCS-Beispiel-AMIs und mehreren Netzwerkschnittstellen konfiguriert sind, funktionieren derzeit nicht, wenn die Subnetze nur für die Verwendung von IPv6 konfiguriert sind. Verwenden Sie stattdessen Dual-Stack-Subnetze (IPv4 und IPv6) oder Subnetze. IPv4-only

Finden Sie aktuelle AWS PCS-Beispiel-AMIs

AWS-Managementkonsole

Für AWS PCS-Beispiel-AMIs gilt die folgende Namenskonvention:

```
aws-pcs-sample_ami-OS-architecture
```

Akzeptierte Werte

- *OS* – al2023
- *architecture* – x86_64 oder arm64

Ein Beispiel-AMI unterstützt jede Slurm-Version, die nicht EOL ist. Der AMI-Name identifiziert keine Slurm-Version. Stattdessen listet die AMI-Beschreibung die Versionen auf, die das AMI unterstützt, zum Beispiel Slurm: 25.05, 25.11, 26.05.

> **Note**
>
> Beispiel-AMIs, die vor der Version Slurm 26.05 veröffentlicht wurden, unterstützen eine einzelne Slurm-Version und enthalten beispielsweise den Scheduler und seine Hauptversion im Namen. aws-pcs-sample_ami-al2023-arm64-slurm-25.11 Diese AMIs bleiben für Slurm 25.11 und frühere Versionen verfügbar.
> Die Beispiel-AMIs mit einer einzigen Version für Slurm 25.05 verwenden Amazon Linux (2amzn2) anstelle von Amazon Linux 2023 (). al2023

Um zu finden AWS PCS-Beispiel-AMIs

1. Öffnen Sie die Amazon EC2-Konsole.
2. Navigieren Sie zu AMIs.
3. Wählen Sie Öffentliche Abbilder aus.
4. Suchen Sie unter AMI anhand eines Attributs oder Tags suchen Sie anhand des Vorlagennamens nach einem AMI.

   Beispiele

   - Beispiel-AMI für Arm64-Instances

     ```
     aws-pcs-sample_ami-al2023-arm64
     ```

   - Beispiel-AMI für x86-Instances

     ```
     aws-pcs-sample_ami-al2023-x86_64
     ```

> **Note**
>
> Wenn mehrere AMIs vorhanden sind, verwenden Sie das AMI mit dem neuesten Zeitstempel.

> **Note**
>
> Überprüfen Sie die AMI-Beschreibung, um sicherzustellen, dass das AMI die Slurm-Version Ihres Clusters unterstützt.

5. Verwenden Sie die AMI-ID, wenn Sie eine Rechenknotengruppe erstellen oder aktualisieren.

AWS CLI

Das neueste AWS PCS-Beispiel-AMI finden Sie mit den folgenden Befehlen. *region-code*Ersetzen Sie es durch die Version, AWS-Region in der Sie AWS PCS verwendenus-east-1, z. B., und *slurm-version* durch die Slurm-Hauptversion Ihres Clusters, z. B. 26.05 Der Beschreibungsfilter wählt ein AMI aus, das diese Version unterstützt.

- x86_64

```
aws ec2 describe-images --region region-code --owners amazon \
--filters 'Name=name,Values=aws-pcs-sample_ami-al2023-x86_64*' \
           'Name=description,Values=*slurm-version*' \
           'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId,Description]' --output
 text
```

- Arm 64

```
aws ec2 describe-images --region region-code --owners amazon \
--filters 'Name=name,Values=aws-pcs-sample_ami-al2023-arm64*' \
           'Name=description,Values=*slurm-version*' \
           'Name=state,Values=available' \
--query 'sort_by(Images, &CreationDate)[-1].[Name,ImageId,Description]' --output
 text
```

Verwenden Sie die AMI-ID, wenn Sie eine Rechenknotengruppe erstellen oder aktualisieren.

Erfahren Sie mehr über AWS PCS-Beispiel-AMIs

Den Inhalt und die Konfigurationsdetails für aktuelle und frühere Versionen der AWS PCS-Beispiel-AMIs finden Sie unterVersionshinweise für AWS PCS-Beispiel-AMIs.

Erstellen Sie Ihre eigenen AMIs, die kompatibel sind mit AWS 5 STÜCK

Informationen zum Erstellen eigener AMIs, die mit AWS PCS funktionieren, finden Sie unterBenutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS

AWS PCS ist so konzipiert, dass es mit Amazon Machine Images (AMI) funktioniert, die Sie für den Service bereitstellen. Auf diesen AMIs können beliebige Software und Konfigurationen installiert sein, sofern auf ihnen der AWS PCS-Agent und eine kompatible Version von Slurm korrekt installiert und konfiguriert sind. Sie müssen die von Ihnen AWS bereitgestellten Installationsprogramme verwenden, um die AWS PCS-Software auf Ihrem benutzerdefinierten AMI zu installieren. Wir empfehlen Ihnen, AWS zur Installation von Slurm auf Ihrem benutzerdefinierten AMI bereitgestellte Installationsprogramme zu verwenden, aber Sie können Slurm auch selbst installieren, wenn Sie dies bevorzugen (nicht empfohlen).

> **Note**
>
> Wenn Sie AWS PCS ausprobieren möchten, ohne ein benutzerdefiniertes AMI zu erstellen, können Sie ein Beispiel-AMI verwenden, das von bereitgestellt wird AWS. Weitere Informationen finden Sie unter Verwenden von Amazon Machine Images (AMIs) als Beispiel mit AWS STCK..

> **Important**
>
> AWS PCS benötigt derzeit einen Kernel mit IPv4 Unterstützung für lokale Knotenkommunikation, auch wenn Sie AWS PCS in einem IPv6 reinen Netzwerk verwenden.

Dieses Tutorial hilft Ihnen bei der Erstellung eines AMI, das mit PCS-Rechenknotengruppen verwendet werden kann, um Ihr HPC und Ihre AI/ML Workloads zu unterstützen.

Themen

- Schritt 1 — Eine temporäre Instanz starten
- Schritt 2 — Installieren Sie den AWS PCS-Agent
- Schritt 3 — Slurm installieren
- Schritt 4 — (Optional) Zusätzliche Treiber, Bibliotheken und Anwendungssoftware installieren
- Schritt 5 — Erstellen Sie ein mit AWS PCS kompatibles AMI
- Schritt 6 — Verwenden Sie das benutzerdefinierte AMI mit einer AWS PCS-Compute-Knotengruppe
- Schritt 7 — Beenden Sie die temporäre Instanz

Schritt 1 — Eine temporäre Instanz starten

Starten Sie eine temporäre Instanz, mit der Sie die AWS PCS-Software und den Slurm-Scheduler installieren und konfigurieren können. Sie verwenden diese Instance, um ein mit AWS PCS kompatibles AMI zu erstellen.

So starten Sie eine temporäre Instance

1. Öffnen Sie die Amazon EC2-Konsole.
2. Wählen Sie im Navigationsbereich Instances und anschließend Launch Instances aus, um den Assistenten zum Starten neuer Instances zu öffnen.
3. (Optional) Geben Sie im Abschnitt Name und Tags einen Namen für die Instance ein, z. `PCS-AMI-instance` B. Der Name wird der Instance als Ressourcen-Tag (`Name=PCS-AMI-instance`) zugewiesen.
4. Wählen Sie im Bereich Application and OS Images (Anwendungs- und Betriebssystem-Images) ein AMI für eines der unterstützten Betriebssysteme aus.
5. Wählen Sie im Bereich Instance type (Instance-Typ) einen supported instance type (unterstützten Instance-Typ) aus.
6. Wählen Sie im Bereich Key pair (Schlüsselpaar) das Schlüsselpaar aus, das für die Instance verwendet werden soll.
7. Gehen Sie im Abschnitt Netzwerkeinstellungen wie folgt vor:

- Wählen Sie für Firewall (Sicherheitsgruppen) die Option Bestehende Sicherheitsgruppe auswählen und anschließend eine Sicherheitsgruppe aus, die eingehenden SSH-Zugriff auf Ihre Instance ermöglicht.

8. Konfigurieren Sie im Bereich Storage (Speicher) die Volumes nach Bedarf. Stellen Sie sicher, dass ausreichend Speicherplatz für die Installation Ihrer eigenen Anwendungen und Bibliotheken konfiguriert ist.
9. Wählen Sie in der Übersicht Launch instance (Instance starten) aus.

Schritt 2 — Installieren Sie den AWS PCS-Agent

Installieren Sie den Agenten, der die von AWS PCS gestarteten Instanzen für die Verwendung mit Slurm konfiguriert. Weitere Informationen zum AWS PCS-Agenten finden Sie unter. AWS PCS-Agent-Versionen

Um den zu installieren AWS PCS-Agent

1. Stellen Sie eine Verbindung zu der Instance her, die Sie gestartet haben. Weitere Informationen finden Sie unter Stellen Sie eine Verbindung zu Ihrer Linux-Instance her.
2. (Optional) Um sicherzustellen, dass alle Ihre Softwarepakete auf dem neuesten Stand sind, führen Sie ein schnelles Software-Update auf Ihrer Instance durch. Dieser Vorgang kann einige Minuten dauern.

   - Amazon Linux 2, Amazon Linux 2023, RHEL 9, RHEL 8, Rocky Linux 9 und Rocky Linux 8

     ```
     sudo yum update -y
     ```

   - Ubuntu 22.04 und Ubuntu 24.04

     ```
     sudo apt-get update && sudo apt-get upgrade -y
     ```

3. Starten Sie die Instance neu und stellen Sie die Verbindung zur Instance wieder her.
4. Laden Sie die Installationsdateien des AWS PCS-Agenten herunter. Die Installationsdateien sind in eine komprimierte Tarball (`.tar.gz`) -Datei gepackt. Laden Sie die neueste stabile Version mit dem folgenden Befehl herunter. Ersetzen Sie es *`us-east-1`* durch den AWS-Region Ort, an dem Sie Ihre temporäre Instance gestartet haben.

```
curl https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.5.1-1.tar.gz -o aws-pcs-agent-v1.5.1-1.tar.gz
```

Sie können auch die neueste Version abrufen, indem Sie die Versionsnummer durch `latest` den vorherigen Befehl ersetzen (zum Beispiel:`aws-pcs-agent-v1-latest.tar.gz`).

> **Note**
>
> Dies könnte sich in zukünftigen Versionen der AWS PCS-Agentensoftware ändern.

5. (Optional) Überprüfen Sie die Authentizität und Integrität des AWS PCS-Software-Tarballs. Diese Vorgehensweise wird empfohlen, um die Identität des Software-Publishers zu überprüfen und sicherzustellen, dass die Datei seit ihrer Veröffentlichung nicht verändert oder beschädigt wurde.

   a. Laden Sie den öffentlichen GPG-Schlüssel für AWS PCS herunter und importieren Sie ihn in Ihren Schlüsselbund. Ersetzen Sie ihn *us-east-1* durch den AWS-Region Ort, an dem Sie Ihre temporäre Instance gestartet haben. Der Befehl sollte einen Schlüsselwert zurückgeben. Notieren Sie sich den Schlüsselwert; Sie verwenden ihn im nächsten Schritt.

   ```
   wget https://aws-pcs-repo-public-keys-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-public-key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. Führen Sie den folgenden Befehl aus, um den Fingerabdruck des GPG-Schlüssels zu überprüfen.

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   Der Befehl sollte einen Fingerabdruck zurückgeben, der mit dem folgenden identisch ist:

   ```
   1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   ```

   > **Important**
   >
   > Führen Sie das Installationsskript für den AWS PCS-Agenten nicht aus, wenn der Fingerabdruck nicht übereinstimmt. AWS Support kontaktieren.

c. Laden Sie die Signaturdatei herunter und überprüfen Sie die Signatur der AWS PCS-Software-Tarball-Datei. *us-east-1*Ersetzen Sie durch den AWS-Region Ort, an dem Sie Ihre temporäre Instance gestartet haben.

```
wget https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/
aws-pcs-agent-v1.5.1-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-agent-v1.5.1-1.tar.gz.sig
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
gpg: assuming signed data in './aws-pcs-agent-v1.5.1-1.tar.gz'
gpg: Signature made Thu 06 Nov 2025 11:10:36 AM CET using RSA key ID ECC0AE5C
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   Subkey fingerprint: B7E1 8788 3517 6A74 C3D5  EAF5 6088 136D ECC0 AE5C
```

Wenn das Ergebnis Folgendes enthält `Good signature` und der Fingerabdruck mit dem im vorherigen Schritt zurückgegebenen Fingerabdruck übereinstimmt, fahren Sie mit dem nächsten Schritt fort.

> **Important**
>
> Führen Sie das AWS PCS-Softwareinstallationsskript nicht aus, wenn der Fingerabdruck nicht übereinstimmt. AWS Support kontaktieren.

6. Extrahieren Sie die Dateien aus der komprimierten `.tar.gz` Datei und navigieren Sie zum entpackten Verzeichnis.

```
tar -xf aws-pcs-agent-v1.5.1-1.tar.gz && \
    cd aws-pcs-agent
```

7. Installieren Sie die AWS PCS-Software.

```
sudo ./installer.sh
```

8. Überprüfen Sie die AWS PCS-Softwareversionsdatei, um eine erfolgreiche Installation zu bestätigen.

```
cat /opt/aws/pcs/version
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
AGENT_INSTALL_DATE='Fri Dec 13 12:28:43 UTC 2024'
AGENT_VERSION='1.5.1'
AGENT_RELEASE='1'
```

Schritt 3 — Slurm installieren

Installieren Sie eine Version von Slurm, die mit AWS PCS kompatibel ist. Weitere Informationen finden Sie unter Slurm-Versionen in AWS 5 STCK.

> **Note**
>
> Wenn Sie ein AMI haben, auf dem eine frühere Version der Slurm-Software installiert ist, müssen Sie die folgenden Schritte ausführen, um die neue Version von Slurm zu installieren. Der AWS PCS-Agent aktiviert zur Laufzeit die richtige Version der Slurm-Binärdateien, entsprechend der Slurm-Version, die bei der Cluster-Erstellung konfiguriert wurde.

Um Slurm zu installieren

1. Stellen Sie eine Verbindung zu derselben temporären Instanz her, auf der Sie die AWS PCS-Software installiert haben.
2. Laden Sie die Slurm-Installationssoftware herunter. Das Slurm-Installationsprogramm ist in eine komprimierte Tarball () `.tar.gz` -Datei gepackt. Laden Sie die neueste stabile Version mit dem folgenden Befehl herunter. Ersetzen Sie es *`us-east-1`* durch die AWS-Region Ihrer temporären Instanz.

   ```
   curl https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz \
        -o aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz
   ```

   Sie können auch die neueste Version abrufen, indem Sie die Versionsnummer durch `latest` den vorherigen Befehl ersetzen (zum Beispiel:`aws-pcs-slurm-26.05-installer-`

`latest.tar.gz`). Eine vollständige Liste der verfügbaren Versionen mit Prüfsummen finden Sie unterÜberprüfen Sie die Installationsprogramme anhand einer Prüfsumme.

> **Note**
>
> Dies könnte sich in zukünftigen Versionen der Slurm-Installationssoftware ändern.

3. (Optional) Überprüfen Sie die Authentizität und Integrität des Slurm-Installations-Tarballs. Diese Vorgehensweise wird empfohlen, um die Identität des Software-Publishers zu überprüfen und sicherzustellen, dass die Datei seit ihrer Veröffentlichung nicht verändert oder beschädigt wurde.

   a. Laden Sie den öffentlichen GPG-Schlüssel für AWS PCS herunter und importieren Sie ihn in Ihren Schlüsselbund. Ersetzen Sie ihn *`us-east-1`* durch den AWS-Region Ort, an dem Sie Ihre temporäre Instance gestartet haben. Der Befehl sollte einen Schlüsselwert zurückgeben. Notieren Sie sich den Schlüsselwert; Sie verwenden ihn im nächsten Schritt.

   ```
   wget https://aws-pcs-repo-public-keys-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-public-key.pub && \
       gpg --import aws-pcs-public-key.pub
   ```

   b. Führen Sie den folgenden Befehl aus, um den Fingerabdruck des GPG-Schlüssels zu überprüfen.

   ```
   gpg --fingerprint 7EEF030EDDF5C21C
   ```

   Der Befehl sollte einen Fingerabdruck zurückgeben, der mit dem folgenden identisch ist:

   ```
   1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
   ```

   > **Important**
   >
   > Führen Sie das Slurm-Installationsskript nicht aus, wenn der Fingerabdruck nicht übereinstimmt. AWS Support kontaktieren.

   c. Laden Sie die Signaturdatei herunter und überprüfen Sie die Signatur der Tarball-Datei des Slurm-Installationsprogramms. *`us-east-1`*Ersetzen Sie durch den AWS-Region Ort, an dem Sie Ihre temporäre Instanz gestartet haben.

```
wget https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/
aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz.sig && \
    gpg --verify ./aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz.sig
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
gpg: assuming signed data in './aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz'
gpg: Signature made Thu 03 Sep 2026 09:05:02 AM UTC using RSA key ID ECC0AE5C
gpg: Good signature from "AWS PCS Packages (AWS PCS Packages)"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1C24 32C1 862F 64D1 F90A  239A 7EEF 030E DDF5 C21C
    Subkey fingerprint: B7E1 8788 3517 6A74 C3D5  EAF5 6088 136D ECC0 AE5C
```

Wenn das Ergebnis Folgendes enthält `Good signature` und der Fingerabdruck mit dem im vorherigen Schritt zurückgegebenen Fingerabdruck übereinstimmt, fahren Sie mit dem nächsten Schritt fort.

> **Important**
>
> Führen Sie das Slurm-Installationsskript nicht aus, wenn der Fingerabdruck nicht übereinstimmt. AWS Support kontaktieren.

4. Extrahieren Sie die Daten aus der komprimierten `.tar.gz`-Datei und wechseln Sie in das extrahierte Verzeichnis.

```
tar -xf aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz && \
    cd aws-pcs-slurm-26.05-installer
```

5. Installiere Slurm. Das Installationsprogramm lädt Slurm und seine Abhängigkeiten herunter, kompiliert und installiert es. Es dauert mehrere Minuten, abhängig von den Spezifikationen der temporären Instanz, die Sie ausgewählt haben.

```
sudo ./installer.sh -y
```

6. Überprüfen Sie die Scheduler-Versionsdatei, um die Installation zu bestätigen.

```
cat /opt/aws/pcs/scheduler/slurm-26.05/version
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
SLURM_INSTALL_DATE='Thu Sep  3 15:15:37 UTC 2026'
SLURM_VERSION='26.05.4'
PCS_SLURM_RELEASE='1'
```

Schritt 4 — (Optional) Zusätzliche Treiber, Bibliotheken und Anwendungssoftware installieren

Installieren Sie zusätzliche Treiber, Bibliotheken und Anwendungssoftware auf der temporären Instanz. Die Installationsverfahren variieren je nach den spezifischen Anwendungen und Bibliotheken. Wenn Sie noch kein benutzerdefiniertes AMI für AWS PCS erstellt haben, empfehlen wir Ihnen, zunächst ein AMI nur mit der AWS PCS-Software und installiertem Slurm zu erstellen und zu testen und dann schrittweise Ihre eigene Software und Konfigurationen hinzuzufügen, sobald Sie den ersten Erfolg bestätigt haben.

Beispiele

- Software für Elastic Fabric Adapter (EFA). Weitere Informationen finden Sie unter Erste Schritte mit EFA und MPI für HPC-Workloads auf Amazon EC2 im Amazon Elastic Compute Cloud-Benutzerhandbuch.
- Client für Amazon Elastic File System (Amazon EFS). Weitere Informationen finden Sie unter Manuelles Installieren des Amazon EFS-Clients im Amazon Elastic File System-Benutzerhandbuch.
- Lustre-Client, um Amazon FSx for Lustre und Amazon File Cache zu verwenden. Weitere Informationen finden Sie unter Installation des Lustre-Clients im FSx for Lustre-Benutzerhandbuch.
- CloudWatch Amazon-Agent, um CloudWatch Logs and Metrics zu verwenden. Weitere Informationen finden Sie unter Installieren des CloudWatch Agenten im CloudWatch Amazon-Benutzerhandbuch.
- AWS Neuron, um die Instance-Typen trn* und inf* zu verwenden. Weitere Informationen finden Sie in der Neuron-Dokumentation.AWS
- NVIDIA-Treiber, CUDA und DCGM, um die Instanztypen p* oder g* zu verwenden.

Schritt 5 — Erstellen Sie ein mit AWS PCS kompatibles AMI

Nachdem Sie die erforderlichen Softwarekomponenten installiert haben, erstellen Sie ein AMI, das Sie wiederverwenden können, um Instances in AWS PCS-Compute-Knotengruppen zu starten.

> **Important**
>
> AWS PCS benötigt derzeit einen Kernel mit IPv4 Unterstützung für lokale Knotenkommunikation, auch wenn Sie AWS PCS in einem IPv6 reinen Netzwerk verwenden.

So erstellen Sie ein AMI aus Ihrer temporären Instance:

1. Öffnen Sie die Amazon EC2-Konsole.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die temporäre Instanz aus, die Sie erstellt haben. Wählen Sie Aktionen, Image, Image erstellen.
4. Gehen Sie bei Create Image (Image erstellen) wie folgt vor:
   a. Geben Sie unter Image name (Image-Name) einen beschreibenden Namen für das AMI ein.
   b. (Optional:) Geben Sie bei Image description (Image-Beschreibung) eine kurze Beschreibung des Zwecks des AMI ein.
   c. Wählen Sie Create Image (Image erstellen) aus.
5. Wählen Sie im Navigationsbereich AMIs aus.
6. Suchen Sie das AMI, das Sie erstellt haben, in der Liste. Warten Sie, bis sich der Status von Ausstehend auf Verfügbar ändert, und verwenden Sie es dann mit einer AWS PCS-Compute-Knotengruppe.

Schritt 6 — Verwenden Sie das benutzerdefinierte AMI mit einer AWS PCS-Compute-Knotengruppe

Sie können Ihr benutzerdefiniertes AMI mit einer neuen oder vorhandenen AWS PCS-Compute-Knotengruppe verwenden.

> **Important**
>
> AWS PCS benötigt derzeit einen Kernel mit IPv4 Unterstützung für lokale Knotenkommunikation, auch wenn Sie AWS PCS in einem IPv6 reinen Netzwerk verwenden.

New compute node group

Um das benutzerdefinierte AMI zu verwenden

1. Öffnen Sie die AWS PCS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, in dem Sie das benutzerdefinierte AMI verwenden möchten, und wählen Sie dann Compute Node Groups aus.
4. Erstellen Sie eine neue Compute-Knotengruppe. Weitere Informationen finden Sie unter Erstellen einer Compute-Knotengruppe in AWS STCK. Suchen Sie unter AMI-ID nach dem Namen oder der ID des benutzerdefinierten AMI, das Sie verwenden möchten. Schließen Sie die Konfiguration der Compute-Knotengruppe ab und wählen Sie dann Create Compute Node Group aus.
5. (Optional) Vergewissern Sie sich, dass das AMI Instance-Starts unterstützt. Starten Sie eine Instance in der Compute-Knotengruppe. Sie können dies tun, indem Sie die Compute-Knotengruppe so konfigurieren, dass sie über eine einzelne statische Instanz verfügt, oder Sie können einen Job an eine Warteschlange senden, die die Compute-Knotengruppe verwendet.

    a. Überprüfen Sie die Amazon EC2 EC2-Konsole, bis eine Instance angezeigt wird, die mit der neuen Compute-Knotengruppen-ID gekennzeichnet ist. Weitere Informationen dazu finden Sie unter.. Suchen nach Rechenknotengruppeninstanzen in AWS PCS

    b. Wenn Sie sehen, dass eine Instance gestartet wird und ihr Bootstrap-Vorgang abgeschlossen ist, vergewissern Sie sich, dass sie das erwartete AMI verwendet. Wählen Sie dazu die Instance aus und überprüfen Sie dann die AMI-ID unter Details. Es sollte dem AMI entsprechen, das Sie in den Einstellungen der Compute-Knotengruppe konfiguriert haben.

    c. (Optional) Aktualisieren Sie die Skalierungskonfiguration für die Compute-Knotengruppe auf Ihre bevorzugten Werte.

Existing compute node group

Um das benutzerdefinierte AMI zu verwenden

1. Öffnen Sie die AWS PCS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, in dem Sie das benutzerdefinierte AMI verwenden möchten, und wählen Sie dann Compute Node Groups aus.
4. Wählen Sie die Knotengruppe aus, die Sie konfigurieren möchten, und klicken Sie auf Bearbeiten. Suchen Sie unter AMI-ID nach dem Namen oder der ID des benutzerdefinierten AMI, das Sie verwenden möchten. Beenden Sie die Konfiguration der Compute-Knotengruppe und wählen Sie dann Update. Neue Instances, die in der Compute-Knotengruppe gestartet werden, verwenden die aktualisierte AMI-ID. Bestehende Instances werden weiterhin das alte AMI verwenden, bis AWS PCS sie ersetzt. Weitere Informationen finden Sie unter Aktualisierung einer AWS PCS-Rechenknotengruppe.
5. (Optional) Vergewissern Sie sich, dass das AMI Instance-Starts unterstützt. Starten Sie eine Instance in der Compute-Knotengruppe. Sie können dies tun, indem Sie die Compute-Knotengruppe so konfigurieren, dass sie über eine einzelne statische Instanz verfügt, oder Sie können einen Job an eine Warteschlange senden, die die Compute-Knotengruppe verwendet.

    a. Überprüfen Sie die Amazon EC2 EC2-Konsole, bis eine Instance angezeigt wird, die mit der neuen Compute-Knotengruppen-ID gekennzeichnet ist. Weitere Informationen dazu finden Sie unter.. Suchen nach Rechenknotengruppeninstanzen in AWS PCS

    b. Wenn Sie sehen, dass eine Instance gestartet wird und ihr Bootstrap-Vorgang abgeschlossen ist, vergewissern Sie sich, dass sie das erwartete AMI verwendet. Wählen Sie dazu die Instance aus und überprüfen Sie dann die AMI-ID unter Details. Es sollte dem AMI entsprechen, das Sie in den Einstellungen der Compute-Knotengruppe konfiguriert haben.

    c. (Optional) Aktualisieren Sie die Skalierungskonfiguration für die Compute-Knotengruppe auf Ihre bevorzugten Werte.

Schritt 7 — Beenden Sie die temporäre Instanz

Nachdem Sie bestätigt haben, dass Ihr AMI wie vorgesehen mit AWS PCS funktioniert, können Sie die temporäre Instance beenden, damit keine Gebühren mehr dafür anfallen.

So beenden Sie die temporäre Instance:

1. Öffnen Sie die Amazon EC2-Konsole.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie die temporäre Instance aus, die Sie erstellt haben, und wählen Sie Actions, Instance state, Terminate Instance aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Terminate.

Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK

AWS stellt eine herunterladbare Datei bereit, mit der die AWS PCS-Software auf einer Instance installiert werden kann. AWS bietet auch Software, mit der relevante Versionen von Slurm und seinen Abhängigkeiten heruntergeladen, kompiliert und installiert werden können. Sie können diese Anweisungen verwenden, um benutzerdefinierte AMIs für die Verwendung mit AWS PCS zu erstellen, oder Sie können Ihre eigenen Methoden verwenden.

Inhalt

- AWS Installationsprogramm für die PCS-Agent-Software
- Slurm-Installationsprogramm
- Unterstützte Betriebssysteme
- Unterstützte Instance-Typen
- Unterstützte Slurm-Versionen
- Überprüfen Sie die Installationsprogramme anhand einer Prüfsumme

AWS Installationsprogramm für die PCS-Agent-Software

Das Installationsprogramm der AWS PCS-Agentsoftware konfiguriert eine Instanz so, dass sie während des Instanz-Bootstrap-Vorgangs mit AWS PCS funktioniert. Sie müssen die von uns AWS bereitgestellten Installationsprogramme verwenden, um den AWS PCS-Agenten auf Ihrem benutzerdefinierten AMI zu installieren.

Weitere Informationen zur AWS PCS-Agentensoftware finden Sie unter. AWS PCS-Agent-Versionen

Slurm-Installationsprogramm

Das Slurm-Installationsprogramm lädt relevante Versionen von Slurm und seinen Abhängigkeiten herunter, kompiliert und installiert sie. Sie können den Slurm-Installer verwenden, um benutzerdefinierte AMIs für PCS zu erstellen. AWS Sie können auch Ihre eigenen Mechanismen verwenden, wenn diese mit der Softwarekonfiguration übereinstimmen, die der Slurm-Installer bereitstellt. Weitere Informationen zur AWS PCS-Unterstützung für Slurm finden Sie unter. Slurm-Versionen in AWS 5 STCK

Die AWS mitgelieferte Software installiert Folgendes:

- Slurm in der angeforderten Haupt- und Wartungsversion (aktuell Version 26.05.x) - Lizenz GPL 2 https://github.com/SchedMD/slurm?tab=License-1-ov-file
  - Slurm ist gebaut mit set to `--sysconfdir /etc/slurm`
  - Slurm ist mit der Option gebaut und `--enable-pam --without-munge`
  - Slurm ist mit der Option gebaut `--sharedstatedir=/run/slurm/`
  - Slurm ist mit PMIX- und JWT-Unterstützung ausgestattet
  - Slurm ist zum Beispiel unter installiert `/opt/aws/pcs/scheduler/slurm-`*`version`* `/opt/aws/pcs/scheduler/slurm-26.05`
- OpenPMix (Version 4.2.6) — Lizenz https://github.com/openpmix/openpmix?tab=License-1-ov-file
  - OpenPMix wird als Unterverzeichnis von installiert `/opt/aws/pcs/scheduler/`
- libjwt (Version 1.17.0) — Lizenz MPL-2.0
  - libjwt wird als Unterverzeichnis von installiert `/opt/aws/pcs/scheduler/`

Die AWS mitgelieferte Software ändert die Systemkonfiguration wie folgt:

- Die vom Build erstellte `systemd` Slurm-Datei wird beispielsweise `/etc/systemd/system/` mit dem Dateinamen `slurmd-`*`version`*`.service` kopiert. `slurmd-26.05.service`
- Wenn sie nicht existieren, werden ein Slurm-Benutzer und eine Gruppe (`slurm:slurm`) mit UID/GID of erstellt. `401`
- Der Ordner `/etc/aws/pcs/scheduler/slurm-`*`version`*`/plugstack.conf.d/` wird erstellt, um Ihre Erweitern Sie die Slurm-Funktionalität auf AWS PCS mit SPANK-Plugins Konfiguration zu speichern.
- Auf Amazon Linux 2 und Rocky Linux 9 fügt die Installation das EPEL Repository hinzu, um die erforderliche Software zum Erstellen von Slurm oder seinen Abhängigkeiten zu installieren.

- Auf RHEL9 wird die Installation aktiviert `codeready-builder-for-rhel-9-rhui-rpms` und `epel-release-latest-9` von, um die erforderliche Software `fedoraproject` zum Erstellen von Slurm oder seinen Abhängigkeiten zu installieren.

Unterstützte Betriebssysteme

Siehe Unterstützte Betriebssysteme in AWS PCS.

> **Note**
>
> AWS Deep Learning AMIs (DLAMI) -Versionen, die auf Amazon Linux 2023 und Ubuntu 22.04 basieren, sollten mit der PCS-Software und den AWS Slurm-Installern kompatibel sein. Weitere Informationen finden Sie unter Choosing Your https://docs.aws.amazon.com/dlami/latest/devguide/options.html DLAMI im Developer Guide. AWS Deep Learning AMIs

Unterstützte Instance-Typen

AWS Die PCS-Software und die Slurm-Installationsprogramme unterstützen jeden x86_64- oder arm64-Instance-Typ, auf dem eines der unterstützten Betriebssysteme ausgeführt werden kann.

Unterstützte Slurm-Versionen

Siehe Slurm-Versionen in AWS 5 STCK.

Überprüfen Sie die Installationsprogramme anhand einer Prüfsumme

Sie können SHA256-Prüfsummen verwenden, um die Tarball-Dateien des Installationsprogramms (.tar.gz) zu überprüfen. Diese Vorgehensweise wird empfohlen, um die Identität des Software-Publishers zu überprüfen und zu prüfen, ob die Anwendung seit der Veröffentlichung nicht verändert oder beschädigt wurde.

Um einen Tarball zu verifizieren

Verwenden Sie das Hilfsprogramm sha256sum für die SHA256-Prüfsumme und geben Sie den Tarball-Dateinamen an. Sie müssen den Befehl in dem Verzeichnis ausführen, in dem Sie die Tarball-Datei gespeichert haben.

- SHA256

```
$ sha256sum tarball_filename.tar.gz
```

Der Befehl sollte einen Prüfsummenwert im folgenden Format zurückgeben.

```
checksum_value tarball_filename.tar.gz
```

Vergleichen Sie den vom Befehl zurückgegebenen Prüfsummenwert mit dem in der folgenden Tabelle angegebenen Prüfsummenwert. Wenn die Prüfsummen übereinstimmen, ist es sicher, das Installationsskript auszuführen.

> **Important**
>
> Wenn die Prüfsummen nicht übereinstimmen, führen Sie das Installationsskript nicht aus. Wenden Sie sich an Support.

Zum Beispiel generiert der folgende Befehl die SHA256-Prüfsumme für den Slurm 26.05.4-1-Tarball:

```
$ sha256sum aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz
```

Beispielausgabe:

```
e5e618fd1a4433878de417e28179785e117874bdde0e2a5da8fee344eeb1a94d aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz
```

In den folgenden Tabellen sind die Prüfsummen für aktuelle Versionen der Installationsprogramme aufgeführt. Ersetzen Sie *us-east-1* durch den AWS-Region Ort, an dem Sie PCS verwenden AWS .

AWS PCS-Agent

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
AWS PCS-Agent 1.5.1-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/`	`9ac3fff38d75774852d947bf6823abc5a75e`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
	aws-pcs-agent/aws-pcs-agent-v1.5.1-1.tar.gz	5c6325cd5aa5a3172b1defa72f9d
AWS PCS-Agent 1.5.0-1	https://aws-pcs-repo-*us-east-1*.s3.*us-east-1*.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.5.0-1.tar.gz	95a0de1a59d569e28f4dfbdf79da02bb49c4f4febaddc18a5e621c3bee09fa11
AWS PCS-Agent 1.4.1-1	https://aws-pcs-repo-*us-east-1*.s3.*us-east-1*.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.4.1-1.tar.gz	a5829029a293a1e7486c5a2c3c84c15c9d581b4918d2639c8c3ac5a96d7f10f5
AWS PCS-Agent 1.4.0-1	https://aws-pcs-repo-*us-east-1*.s3.*us-east-1*.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.4.0-1.tar.gz	e9a342478483df8e4666741ef2aad504321676508d489f88e96311297990a17f
AWS PCS-Agent 1.3.2-1	https://aws-pcs-repo-*us-east-1*.s3.*us-east-1*.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.2-1.tar.gz	06b32a952a1c849e3442e35c28ac2e4d6962b09286cad748f3c83d561b52ec6f

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
AWS PCS-Agent 1.3.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.1-1.tar.gz`	`5b7f1eb7b3a86bd2d331b5cb0138d868dc9452da34b480becd86af892c7e8d19`
AWS PCS-Agent 1.3.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.3.0-1.tar.gz`	`eadc9b65c3db248bdde2a6c41814dfb1b97239f24ad55e03d8526dd9ab4a8d16`
AWS PCS-Agent 1.2.2-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.2-1.tar.gz`	`fd7b6ea5442db75d723fc4971781ce6ae511baa21b87c4286fc1df8127b282b8`
AWS PCS-Agent 1.2.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.1-1.tar.gz`	`2b784643ca01ccca1baa64fbfb34bb41efe8bdca69470998b74ce3962bc271d4`
AWS PCS-Agent 1.2.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.2.0-1.tar.gz`	`470db8c4fc9e50277b6317f98584b6b547e73523043e34f018eecae767846805`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
AWS PCS-Agent 1.1.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.1-1.tar.gz`	`bef078bf60a6d8ecde2e6c49cd34d088703f02550279e3bf483d57a235334dc6`
AWS PCS-Agent 1.1.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.1.0-1.tar.gz`	`594c32194c71bccc5d66e5213213ae38dd2c6d2f9a950bb01accea0bbab0873a`
AWS PCS-Agent 1.0.1-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.1-1.tar.gz`	`04e22264019837e3f42d8346daf5886eaacecd21571742eb505ea8911786bcb2`
AWS PCS-Agent 1.0.0-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-agent/aws-pcs-agent-v1.0.0-1.tar.gz`	`d2d3d68d00c685435c38af471d7e2492dde5ce9eb222d7b6ef0042144b134ce0`

Slurm-Installationsprogramm

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 26.05.4-1	`https://aws-pcs-repo-us-east-1.s3.us-`	`e5e618fd1a4433878de417e28179785e1178`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
	east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-26.05-installer-26.05.4-1.tar.gz	74bdde0e2a5da8fee344eeb1a94d
Slurm 25.11.8-1	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.11-installer-25.11.8-1.tar.gz	70bf017a088223524949ffd5ef0d0b0eea45b22f904bcad1fbc0c9d92ea4a9c9
Slurm 25.11.7-3	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.11-installer-25.11.7-3.tar.gz	906bd06dc1b4036dff7d1af2e49b0495a5e9e86e971020f869d4dc8bf56d7cb6
Slurm 25.11.7-1	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.11-installer-25.11.7-1.tar.gz	901305999c4b572229aade737d66865fcb6b52fe0c9c1f27d03f1ffda3b9cf13
Slurm 25.11.6-2	https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.11-installer-25.11.6-2.tar.gz	553e74598fcc65b532b57bfa9a89820cd00ac8f233aa464c7d194d35540f3ec5

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 25.11.6-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.11-installer-25.11.6-1.tar.gz`	`8eddffd39aa31fef7d4e73106627212ec9ec14ac66d3f37acd8a95406ab3a8af`
Slurm 25.11.2-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.11-installer-25.11.2-1.tar.gz`	`aa063bc01b2ccd84a82402e8b8dbcd8c7401ebd2e0a670c867d77167944d621a`
Slurm 25.05.9-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.9-1.tar.gz`	`d9be2ae01eac14cd023f48787770878530b308e40b35c4ec1da3a2dbd1a372b7`
Slurm 25.05.8-3	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.8-3.tar.gz`	`8836baa5c354156ed8d2613ac4758be900a6c4a084738c01fc6531846722e795`
Slurm 25.05.8-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.8-2.tar.gz`	`844e75d082cfab7866f4e8d9daaef0225f8af5df6fa8d118619baca395079a33`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 25.05.8-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.8-1.tar.gz`	`dc8b3de2588e99985d5148e62d6aa947fe39005f721892223133c5f67f243f73`
Slurm 25.05.7-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.7-1.tar.gz`	`5019436389649ce0cacf04cd1d1adf1a4e46b9291967af7bf5f0a8ac4a49e4f0`
Slurm 25.05.5-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.5-1.tar.gz`	`e7bc84db4e71b8c7174e2f581a31233f839affb5306c76a8adba23204dcc703b`
Slurm 25.05.4-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.4-1.tar.gz`	`3b0f93bce441d4f4f6935175f2c1e81cd961cb923adb416fa6689f5592047a7d`
Slurm 25.05.3-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-25.05-installer-25.05.3-1.tar.gz`	`851bb5815b6700ceb30cc4a3fda204ca8ce362c14528c339908983255a936cf0`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 24.11.7-3	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.7-3.tar.gz`	`85e01bf1ead75476df75c7e9d84760becaae844a377777c72be3479e6f27d688`
Slurm 24.11.7-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.7-2.tar.gz`	`af0ad1b68051b655f03247300ec993ba9035516f653fb04c6026f248685c420d`
Slurm 24.11.7-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.7-1.tar.gz`	`73d75be82c6f88f6e248fd0cc779a5630c62d91ebabdd9cf0f61b1943b6d7d09`
Slurm 24.11.6-2	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.6-2.tar.gz`	`f17cd78e0bc6b9c818b794d9d2685cceabdc73f4fbb12f7566ae5b86a5abc32b`
Slurm 24.11.6-1	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.6-1.tar.gz`	`225de9fc18206f5f65f412effe1fd457614ac97ee9822b3ff804a452b0fae522`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 24.11.5-1	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.11-installer-24.11.5-1.tar.gz`	`593efe4d66bef2f3e46d5a382fb5a32f7a3ca2510bcf1b3c85739f4f951810d5`
Slurm 24.05.8-4 (veraltet)	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-4.tar.gz`	`97906967d194aefca933e453c910015ee3ec47ffe07c47ce7bc6cd3c4c79f438`
Slurm 24.05.8-3 (veraltet)	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-3.tar.gz`	`34e6a5989e9c5cc429bf7cfa7c48fe52b74cfea58fcee50a27ac0d1a81c4d99f`
Slurm 24.05.8-2 (veraltet)	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-2.tar.gz`	`c494b0b55c319a4c2f3faf668c759d46c32c4c7aa94ae97d94128328fe95364b`
Slurm 24.05.8-1 (veraltet)	`https://aws-pcs-repo-us-east-1.s3.us-east-1.amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.8-1.tar.gz`	`210a43b376af082bbad640b2032655885790c5dab0e6489cc327c7310a375849`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 24.05.7-1 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.7-1.tar.gz`	`0b5ed7c81195de2628c78f37c79e63fc4ae99132ca6b019b53a0d68792ee82c5`
Slurm 24.05.5-2 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-24.05-installer-24.05.5-2.tar.gz`	`7cc8d8294f2fbff95fe0602cf9e21e02003b5d96c0730e0a18c6aa04c7a4967b`
Slurm 23.11.10-6 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-6.tar.gz`	`6e3ee3da8c9f4b76e400edfdfcbe0b85d6a597c8ca5fd78291be7106bc4025dd`
Slurm 23.11.10-5 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-5.tar.gz`	`37ba34be8ac0aa28bbf542c26d38a51b200b91185229912e35a6d3c0cbd09165`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 23.11.10-4 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-4.tar.gz`	`bb2d8c919c69dba38d14358f49c7f0427564c5dd4af85a1c9eca2c57ceeae29a`
Slurm 23.11.10-3 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-3.tar.gz`	`488a10ee0fbd57ec0e0ff7ea708a9e3038fafdc025c6bb391c75c2e2a7852a00`
Slurm 23.11.10-2 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-2.tar.gz`	`0bbe85423305c05987931168caf98da08a34c25f9eec0690e8e74de0b7bc8752`
Slurm 23.11.10-1 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.10-1.tar.gz`	`27e8faa9980e92cdfd8cfdc71f937777f0934552ce61e33dac4ecf5a20321e44`

Installer (Installationsprogramm)	URL herunterladen	SHA256-Prüfsumme
Slurm 23.11.9-1 (veraltet)	`https://aws-pcs-repo- us-east-1 .s3.us-east-1 .amazonaws.com/aws-pcs-slurm/aws-pcs-slurm-23.11-installer-23.11.9-1.tar.gz`	`1de7d919c8632fe8e2806611bed4fde1005a4fadc795412456e935c7bba2a9b8`

Versionshinweise für AWS PCS-Beispiel-AMIs

AMIs für die neuesten unterstützten Hauptversionen des Schedulers erhalten Sicherheitsupdates und wichtige Bugfixes. Diese inkrementellen Sicherheitspatches sind nicht in den offiziellen Versionshinweisen enthalten.

Important

Beispiel-AMIs, die sich auf alte Scheduler-Versionen beziehen, werden nicht unterstützt und erhalten keine Updates.

Important

Beispiel-AMIs dienen zu Demonstrationszwecken und werden nicht für Produktionsworkloads empfohlen.

Inhalt

- AWS PCS-Beispiel-AMIs für x86_64
- AWS PCS-Beispiel-AMIs für Arm64

AWS PCS-Beispiel-AMIs für x86_64

Slurm 25.11

AMI-Name

- `aws-pcs-sample_ami-al2023-x86_64-slurm-25.11`

Unterstützte EC2-Instances

- Alle Instanzen mit einem 64-Bit-x86-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach Architectures=x86_64.

AMI-Inhalte

- Unterstützter AWS-Service: AWS PCS
- Betriebssystem: Amazon Linux 2023
- Rechenarchitektur: x86_64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.47.0
- GDR-Kopie: 2.5.1
- NVIDIA-Treiber: 590.48.01
- NVIDIA CUDA: 13.1_590.48.01

Slurm 25.05

AMI-Name

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05`

Unterstützte EC2-Instances

- Alle Instanzen mit einem 64-Bit-x86-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach Architectures=x86_64.

AMI-Inhalte

- Unterstützter AWS-Service: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: x86_64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.43.1
- GDR-Kopie: 2.5.1
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 24.11

> **Note**
>
> AWS PCS unterstützt die Buchhaltung für Slurm 24.11 und höher. Weitere Informationen finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.

AMI-Name

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.11`

Unterstützte EC2-Instances

- Alle Instanzen mit einem 64-Bit-x86-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach`Architectures=x86_64`.

AMI-Inhalte

- Unterstützter AWS Dienst: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: x86_64
- EBS-Volumetyp: gp2

- EFA-Installationsprogramm: 1.33.0
- GDRCopy:: 2.4
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slum 24.05

AMI-Name

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-24.05`

Unterstützte EC2-Instances

- Alle Instanzen mit einem 64-Bit-x86-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach`Architectures=x86_64`.

AMI-Inhalte

- Unterstützter AWS Dienst: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: x86_64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.33.0
- GDRCopy:: 2.4
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 23.11

AMI-Name

- `aws-pcs-sample_ami-amzn2-x86_64-slurm-23.11`

Unterstützte EC2-Instances

- Alle Instanzen mit einem 64-Bit-x86-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach`Architectures=x86_64`.

AMI-Inhalte

- Unterstützter AWS Dienst: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: x86_64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.33.0
- GDRCopy:: 2.4
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

AWS PCS-Beispiel-AMIs für Arm64

Slurm 25.11

AMI-Name

- `aws-pcs-sample_ami-al2023-arm64-slurm-25.11`

Unterstützte EC2-Instances

- Alle Instances mit einem 64-Bit-ARM-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach Architectures=ARM64.

AMI-Inhalte

- Unterstützter AWS-Service: AWS PCS
- Betriebssystem: Amazon Linux 2023

- Rechenarchitektur: arm64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.47.0
- GDR-Kopie: 2.5.1
- NVIDIA-Treiber: 590.48.01
- NVIDIA CUDA: 13.1_590.48.01

Slurm 25.05

AMI-Name

- `aws-pcs-sample_ami-amzn2-arm64-slurm-25.05`

Unterstützte EC2-Instances

- Alle Instances mit einem 64-Bit-ARM-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach Architectures=ARM64.

AMI-Inhalte

- Unterstützter AWS-Service: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: arm64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.43.1
- GDR-Kopie: 2.5.1
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 24.11

> **Note**
>
> AWS PCS unterstützt die Buchhaltung für Slurm 24.11 und höher. Weitere Informationen finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.

AMI-Name

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.11`

Unterstützte EC2-Instances

- Alle Instances mit einem 64-Bit-ARM-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach`Architectures=arm64`.

AMI-Inhalte

- Unterstützter AWS Dienst: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: arm64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.33.0
- GDRCopy:: 2.4
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slum 24.05

AMI-Name

- `aws-pcs-sample_ami-amzn2-arm64-slurm-24.05`

Unterstützte EC2-Instances

- Alle Instances mit einem 64-Bit-ARM-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach`Architectures=arm64`.

AMI-Inhalte

- Unterstützter AWS Dienst: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: arm64
- EBS-Volumetyp: gp2
- EFA-Installationsprogramm: 1.33.0
- GDRCopy:: 2.4
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Slurm 23.11

AMI-Name

- `aws-pcs-sample_ami-amzn2-arm64-slurm-23.11`

Unterstützte EC2-Instances

- Alle Instances mit einem 64-Bit-ARM-Prozessor. Um kompatible Instances zu finden, navigieren Sie zur Amazon EC2 EC2-Konsole. Wählen Sie Instance-Typen und suchen Sie dann nach`Architectures=arm64`.

AMI-Inhalte

- Unterstützter AWS Dienst: AWS PCS
- Betriebssystem: Amazon Linux 2
- Rechenarchitektur: arm64
- EBS-Volumetyp: gp2

- EFA-Installationsprogramm: 1.33.0
- GDRCopy:: 2.4
- NVIDIA-Treiber: 550.127.08
- NVIDIA CUDA: 12.4.1_550.54.15

Unterstützte Betriebssysteme in AWS PCS

AWS PCS verwendet das für eine Rechenknotengruppe konfigurierte Amazon Machine Image (AMI), um EC2-Instances in dieser Rechenknotengruppe zu starten. Das AMI bestimmt das Betriebssystem, das die EC2-Instances verwenden. Sie können das Betriebssystem in AWS PCS-Beispiel-AMIs nicht ändern. Sie müssen ein benutzerdefiniertes AMI erstellen, wenn Sie ein anderes Betriebssystem verwenden möchten. Weitere Informationen finden Sie unter Amazon Machine Images (AMIs) für AWS STK..

Unterstützte Betriebssysteme

- Amazon Linux 2023

  Dies ist das Betriebssystem in den AWS PCS-Beispiel-AMIs.

  > **Important**
  >
  > Beispiel-AMIs dienen zu Demonstrationszwecken und werden nicht für Produktionsworkloads empfohlen. Sie sollten ein benutzerdefiniertes AMI für Produktionsworkloads erstellen und verwenden, auch wenn Sie Amazon Linux 2023 verwenden möchten.

- Amazon Linux 2
- RedHat Linux 9 für Unternehmen (RHEL 9)

  Die On-Demand-Kosten für RHEL sind für jeden Instanztyp höher als für andere unterstützte Betriebssysteme. Weitere Informationen zu den Preisen finden Sie unter On-Demand Preise und Wie wird Red Hat Enterprise Linux auf Amazon Elastic Compute Cloud angeboten und wie wird der Preis berechnet? .

- RedHat Linux 8 für Unternehmen (RHEL 8)
- Rocky Linux 9

  Sie können die offiziellen Rocky Linux 9-AMIs als Basis für ein benutzerdefiniertes AMI verwenden. Ihr benutzerdefinierter AMI-Build schlägt möglicherweise fehl, wenn das Basis-AMI nicht über den neuesten Kernel verfügt.

Um den Kernel zu aktualisieren

1. Starten Sie von hier aus eine Instance mit einer Rocky9-AMI-ID: https://rockylinux.org/cloud-images/
2. Rufen Sie die Instance per SSH auf und führen Sie den folgenden Befehl aus:

```
sudo yum -y update
```

3. Erstellen Sie ein Bild von der Instanz. Sie geben dieses Image als das `ParentImage` für Ihr benutzerdefiniertes AMI an.

- Rocky Linux 8
- Ubuntu 22.04

  Ubuntu 22.04 benötigt sicherere Schlüssel für SSH und unterstützt standardmäßig keine RSA-Schlüssel. Wir empfehlen Ihnen, stattdessen einen ED25519-Schlüssel zu generieren und zu verwenden.
- Ubuntu 24.04

AWS PCS-Agent-Versionen

Die AWS PCS-Agentsoftware konfiguriert die EC2-Instanzen, die AWS PCS für die Verwendung mit Slurm startet. Sie nehmen den Agenten in ein Amazon Machine Images (AMI) auf, das Sie angeben, wenn Sie Rechenknotengruppen für Ihren Cluster erstellen. Die EC2-Instances, die in diesen Rechenknotengruppen gestartet werden, verwenden das angegebene AMI und die darin enthaltene AWS PCS-Agent-Software. Der AWS PCS-Agent ermöglicht es einer EC2-Instance, sich selbst als Teil des Clusters zu registrieren. Um die neueste AWS PCS-Agent-Software verwenden zu können, müssen Sie Ihre benutzerdefinierten AMIs aktualisieren. Weitere Informationen finden Sie unter Schritt 2 — Installieren Sie den AWS PCS-Agent in Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

AWS Version des PCS-Agenten	Datum der Veröffentlichung	Versionshinweise
v1.5.1-1	11. August 2026	• Lösen Sie regionsübergreifende S3-Buckets beim Herunterladen von Node Lifecycle Actions auf. • Ermitteln Sie anhand der `slurmd` Antwort, welche Version gestartet werden soll. RegisterComputeNodeGroupInstance
v1.5.0-1	23. Juli 2026	• Unterstützung für Node-Lifecycle-Aktionen hinzugefügt. Der Agent führt benutzerdefinierte Skripts in den `nodeBootstrapped` `nodeReady` Endphasen des Lebenszyklus eines Rechenknotens aus.
v1.4.1-1	15. Juli 2026	• Es wurde ein symbolischer Link hinzugefügt`/etc/`

AWS Version des PCS-Agenten	Datum der Veröffentlichung	Versionshinweise
		`aws/pcs/scheduler/slurm` , der auf verweist `/etc/aws/pcs/scheduler/slurm-`*`version`*. Der Agent erstellt diesen Link zur Laufzeit auf der Grundlage der gestarteten Version`slurmd`.
v1.4.0-1	7. Mai 2026	• Unterstützung für die Konfiguration und das Starten des `slurmd` Daemons für jede Slurm-Version hinzugefügt, die mit dem Slurm-Controller kompatibel ist, der auf dem PCS-Cluster läuft. AWS
v1.3.2-1	10. März 2026	• Es wurde ein Problem behoben, bei dem Rechenknoten, auf denen RHEL 8.10 oder Rocky Linux 8.10 ausgeführt wurden, aufgrund eines fehlerhaften `curl` SigV4-Backports in diesen Betriebssystemen nicht gestartet werden konnten.

AWS Version des PCS-Agenten	Datum der Veröffentlichung	Versionshinweise
v1.3.1-1	7. November 2025	• Die Deaktivierung von Hyperthreading durch Verwendung des smt/control ``sysfs-Parameters, sofern verfügbar, wurde verbessert. • Es wurde eine mögliche Rennbedingung behoben, bei der die CPU beim Booten gesperrt war, während der PCS-Agent versucht, Hyperthreading zu deaktivieren. • Es wurde ein Problem behoben, das dazu führte, dass die `InstanceType` Felder `InstanceId` und der Slurm-Rechenknoten mit einem Zeitstempel bzw. einem Bindestrich gefüllt wurden.
v1.3.0-1	3. November 2025	• Unterstützung für neue Betriebssysteme hinzugefügt: Amazon Linux 2023, Ubuntu 24, RHEL 8, Rocky 8.

AWS Version des PCS-Agenten	Datum der Veröffentlichung	Versionshinweise
v1.2.2-1	16. Oktober 2025	• Zulässige Abfragen von Instanzmetadaten an einen IPv6-Endpunkt, wenn kein IPv4-Endpunkt verfügbar ist. • Es wurde ein Problem behoben, das verhinderte, dass Hyperthreading deaktiviert wurde, wenn der Kernel gleichgeordnete Threads als CPU-ID-Bereiche zurückgab. • Es wurde ein Problem behoben, das zu falschen Fehlermeldungen in Protokollen führte, wenn Hyperthreading erfolgreich deaktiviert wurde.
v1.2.1-1	19. Juni 2025	• Der AWS PCS-Agent versucht nun für bis zu 30 Minuten, slurmd zu starten, falls der Controller nicht verfügbar ist. • Es wurde ein Problem behoben, das zu einer falschen Slurmd-Konfiguration führte, wenn die Antwort auf einen SLURMDBD-Endpunkt RegisterComputeNodeGroupInstance enthielt.

AWS Version des PCS-Agenten	Datum der Veröffentlichung	Versionshinweise
v1.2.0-1	7. März 2025	• Unterstützung für IPv6 in aktiviert. `slurmd.conf`
v1.1.1-1	13. Dezember 2024	• Ein Problem wurde behoben, bei dem im Call to RegisterComputeNodeGroupInstance eine falsche Slurm-Version gemeldet wurde. • Es wurde ein Problem behoben, bei dem Instanz-Metadaten nicht korrekt abgerufen wurden, wenn ein benutzerdefiniertes Skript ausgeführt `/opt/aws/pcs/etc/bootstrap_hooks/` wurde.
v1.1.0-1	6. Dezember 2024	• Es wurde aktiviert, dass benutzerdefinierte Skripts vor `/opt/aws/pcs/etc/bootstrap_hooks/` den Bootstrap-Schritten ausgeführt werden.
v1.0.1-1	22. Oktober 2024	• Es wurde ein Problem behoben, bei dem NVIDIA-Geräte nicht funktionierten, wenn sie auf GPU-enabled Instances `slurmd` gestartet wurden.
v1.0.0-1	28. August 2024	• Erstversion.

Slurm-Scheduler aktiviert AWS 5 STCK

Slurm ist ein Open-Source-Workload-Manager, der für Linux-Cluster entwickelt wurde und Funktionen zur Auftragsplanung, Ressourcenzuweisung und Auftragsüberwachung für HPC-Workloads bietet. AWS PCS unterstützt den Slurm-Scheduler zur Verwaltung Ihrer Cluster-Workloads.

Topics

- Slurm-Versionen in AWS 5 STCK
- Slurm-Konfiguration in AWS STCK
- Slurm-Buchhaltung in AWS 5 STCK
- Setzen Sie die REST-API ein AWS 5 STCK
- Rechenknoten mit Slurm neu starten AWS 5 STÜCK
- Konfiguration der Hardwaretopologie in AWS STCK
- Konfiguration Multi-Instance GPU (MIG) in AWS STCK
- Erweitern Sie die Slurm-Funktionalität auf AWS PCS mit SPANK-Plugins
- Verwenden Sie die Slurm CLI Filter Plugins, um die Auftragsübermittlung in anzupassen AWS 5 STCK
- Slurm Metriken in AWS STCK
- Erfassen Slurm Metriken mit einem verwalteten Prometheus Kollektor

Slurm-Versionen in AWS 5 STCK

SchedMD erweitert Slurm kontinuierlich mit neuen Funktionen, Optimierungen und Sicherheitspatches. SchedMD veröffentlicht eine neue Hauptversion unter regelmäßige Intervalle und plant, bis zu 3 Versionen gleichzeitig zu unterstützen. AWS PCS ist so konzipiert, dass der Slurm-Controller automatisch mit Patch-Versionen aktualisiert wird.

Wenn SchedMD endet Unterstützung für eine bestimmte Hauptversion, AWS PCS bezeichnet diese Version als End of Life (EOL). Nach EOL können mit dieser Version keine neuen Cluster mehr erstellt werden. Bestehende Cluster können jedoch bis zu 12 Monate ohne garantierten Support weiterlaufen. AWS PCS informiert im Voraus, wenn eine Slurm-Hauptversion kurz vor der EOL-Laufzeit steht, damit Kunden wissen, wann sie ihre Cluster auf eine neuere unterstützte Version aktualisieren müssen.

Wir empfehlen Ihnen, die neueste unterstützte Slurm-Version für die Bereitstellung Ihres Clusters zu verwenden, um auf die neuesten Fortschritte und Verbesserungen zugreifen zu können.

Unterstützte Slurm-Versionen in AWS 5 STCK

Die folgende Tabelle zeigt die unterstützten Slurm-Versionen sowie wichtige Daten und Informationen für jede Version.

Slurm-Version	Veröffentlichungsdatum von SchedMD	AWS PCS-Veröffentlichungsdatum	AWS PCS EOL-Datum	Minimal kompatibel AWS Version des PCS-Agenten	Unterstützt AWS PCS-Beispiel-AMIs
26.05	5/26/2026	9/10/2026	11/30/2027	1,0,0-1	• `aws-pcs-sample_ami-al2023-x86_64` • `aws-pcs-sample_ami-al2023-arm64` Ab 26.05 unterstützt ein einzelnes Beispiel-AMI jede Slurm-Version, die nicht EOL ist, und der AMI-Name enthält keine

Slurm-Version	Veröffentlichungsdatum von SchedMD	AWS PCS-Veröffentlichungsdatum	AWS PCS EOL-Datum	Minimal kompatibel AWS Version des PCS-Agenten	Unterstützt AWS PCS-Beispiel-AMIs
					Slurm-Version mehr. Wie Sie diese AMIs finden, finden Sie unter Verwenden von Beispiel-AMIs.
25,11	11/6/2025	4/9/2026	5/31/2027	1,0,0-1	• `aws-pcs-sample_ami-al2023-x86_64-slurm-25.11` • `aws-pcs-sample_ami-al2023-arm64-slurm-25.11`

Slurm-Version	Veröffentlichungsdatum von SchedMD	AWS PCS-Veröffentlichungsdatum	AWS PCS EOL-Datum	Minimal kompatibel AWS Version des PCS-Agenten	Unterstützt AWS PCS-Beispiel-AMIs
25,05	5/29/2025	10/16/2025	11/30/2026	1,0,0-1	• `aws-pcs-sample_ami-amzn2-x86_64-slurm-25.05` • `aws-pcs-sample_ami-amzn2-arm64-slurm-25.05`

Nicht unterstützte Slurm-Versionen in AWS 5 STCK

Die folgende Tabelle zeigt Slurm-Versionen, die in nicht unterstützt werden AWS 5 STCK.

Version Slurm	Veröffentlichungsdatum von SchedMD	AWS PCS-Veröffentlichungsdatum	AWS PCS EOL-Datum		
24.11	11/29/2024	5/14/2025	5/31/2026		
24,05	5/30/2024	12/18/2024	11/30/2025		
23,11	11/21/2023	8/28/2024	5/31/2025		

Versionshinweise für Slurm-Versionen in AWS 5 STCK

Dieses Thema beschreibt wichtige Änderungen für jede derzeit unterstützte Slurm-Version in AWS STCK. Wir empfehlen Ihnen, die Änderungen zwischen der alten und der neuen Version zu überprüfen, wenn Sie Ihren Cluster aktualisieren.

Slurm 26.05

Änderungen implementiert in AWS STCK

- Sie können konfigurieren `gres.conf` Datensätze für eine Rechenknotengruppe mit `gresCustomSettings` Eigenschaft und deklarieren Sie das planbare GRES mit dem `Gres` benutzerdefinierte Slurm-Einstellung. Auf diese Weise können Sie GPU-Topologie, MIG-Geräte, MPS, Shards und Ihre eigenen generischen Ressourcen konfigurieren. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter GRES-Einstellungen in AWS 5 STCK.
- AWS PCS aktiviert standardmäßig die automatische Slurm-GRES-Erkennung auf Rechenknotengruppen, die über GPUs verfügen. AWS PCS aktiviert die automatische Erkennung nur, wenn das Socket-Layout des Knotens bekannt ist. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter GRES-Einstellungen in AWS 5 STCK.
- AWS PCS konfiguriert standardmäßig die CPU-Topologie einer Rechenknotengruppe. Es bestimmt die Anzahl der NUMA-Domänen anhand des Instanztyps und der folgenden Sätze `Sockets`, `CoresPerSocket`, und `Parameters=numa_node_as_socket` auf jedem Knoten. Weitere Informationen finden Sie unter Konfiguration der Hardwaretopologie in AWS STCK.
- `CommitDelay=1` ist standardmäßig in gesetzt `slurmdbd.conf`. CommitDelay steuert wie lange `slurmdbd` Batches werden in die Buchhaltungsdatenbank übertragen, und die Einstellung verbessert sich `slurmdbd` Leistung. Um einen anderen Wert zu verwenden, setzen Sie `CommitDelay` in der `slurmdbd.conf` benutzerdefinierte Einstellungen für Ihren Cluster.
- Die Slurm-REST-API liest den Benutzernamen aus dem `name` Feld innerhalb der `id` Behauptung eines JWT. Die oberste Ebene `sun`, `username`, und benutzerdefiniert `userclaimfield` Ansprüche sind veraltet. Bestehende Token werden weiterhin authentifiziert. Weitere Informationen finden Sie unter Authentifizierung mit der Slurm-REST-API in AWS 5 STCK.
- Ein einzelnes Beispiel-AMI unterstützt jetzt jede Slurm-Version, die nicht EOL ist, und der AMI-Name enthält keine Slurm-Version mehr. Weitere Informationen finden Sie unter Verwenden von Beispiel-AMIs.
- Cluster, die SPANK-Plugins verwenden, unterstützen fortlaufende Updates, wenn Sie auf 26.05 oder höher aktualisieren. Cluster, die Slurm CLI Filter Plugins verwenden, unterstützen fortlaufende

Updates, wenn Sie von 25.11 oder höher aktualisieren. Weitere Informationen finden Sie unter Aktualisierung der Scheduler-Version eines Clusters in AWS STCK.

- Neue benutzerdefinierte Einstellungen sind verfügbar: `Exclusive` Warteschlangeneinstellung, die `PurgeJobEnvAfter` und `PurgeJobScriptAfter` benutzerdefinierte slurmdbd-Einstellungen, die `PreserveCaseResource` Wert des Slurmdbds `Parameters` Einstellung und die `REBOOT_ONLY` Wert von `HealthCheckNodeState`.

Veraltete Versionen und Löschungen

- Slurm lehnt das ab `ExclusiveUser` Warteschlangeneinstellung und `OverSubscribe=EXCLUSIVE` Wert zugunsten des Neuen `Exclusive` Warteschlangeneinstellung. AWS PCS unterstützt weiterhin `ExclusiveUser` und `OverSubscribe=EXCLUSIVE` am 26.05, aber wir empfehlen Ihnen, zu migrieren `Exclusive`.
- Slurm entfernt `SchedulerParameters=enable_job_state_cache`. Entferne es aus den benutzerdefinierten Slurm-Einstellungen deines Clusters, bevor du auf 26.05 aktualisierst.

Weitere Informationen zu Slurm 26.05 finden Sie in den folgenden Publikationen:

- Versionshinweise zu SchedMD: https://github.com/SchedMD/slurm/blob/slurm-26.05/RELEASE_NOTES.md
- SchedMD-Änderungsprotokoll: https://github.com/SchedMD/slurm/blob/master/CHANGELOG/slurm-26.05.md

Slurm 25.11

Änderungen implementiert in AWS STCK

- Die Auditprotokolle von Scheduler werden jetzt separat über den bereitgestellt `PCS_SCHEDULER_AUDIT_LOGS` Protokolltyp, der die Problembehandlung und Prüfung durch unabhängige Kontrolle der Protokollzustellung vereinfacht. Weitere Informationen finden Sie unter Scheduler-Auditprotokolle in AWS PCS.
- Die beschleunigte Warteschlange ist standardmäßig aktiviert. Jobs, die aufgrund von Knotenproblemen (z. B. aufgrund unzureichender Kapazität) fehlschlagen, können mit der höchsten Planungspriorität in die Warteschlange gestellt werden, indem Sie `sbatch --requeue=expedite`. Dies wird durch die Einstellung `SchedulerParameters=enable_expedited_requeue` gesteuert.

- Das `requeue_delay` Der Parameter ist als benutzerdefinierte Clustereinstellung mit einer Standardeinstellung von 5 Sekunden verfügbar. Bisher war die Verzögerung in der Warteschlange an den Ablauf der Anmeldeinformationen gebunden (70 Sekunden). Administratoren können dies jetzt unabhängig konfigurieren über `SchedulerParameters=requeue_delay=<seconds>`.
- `HealthCheckNodeState` unterstützt jetzt die `START_ONLY` Wert, der das Health-Check-Programm nur beim Start des Knotens ausführt (slurmd start).
- `CommunicationParameters=disable_http` ist standardmäßig so eingestellt, dass HTTP-Endpunkte (Metriken und Integritätstests) deaktiviert werden, die in Slurm 25.11 eingeführt wurden. Um diese Endpunkte wieder zu aktivieren, setzen Sie `CommunicationParameters=enable_http`. Weitere Informationen finden Sie unter Slurm-Metriken in AWS PCS.

Bekannte Probleme

- Slurm 25.11 bestätigt `AllowQOS` und `DenyQOS` Partitionseinstellungen auch wenn `AccountingStorageEnforce=QOS` ist nicht gesetzt. Wenn auf ein QOS verwiesen wird `AllowQOS` oder `DenyQOS` existiert nicht in der Slurm-Buchhaltungsdatenbank, `slurmctld` wird mit einem fatalen Fehler beendet. Stellen Sie sicher, dass alle QOS-Werte in der Partition aufgeführt sind `AllowQOS` und `DenyQOS` Einstellungen sind in der Buchhaltungsdatenbank vorhanden, bevor Slurm 25.11 aktualisiert oder neu gestartet wird.
- Die `slurmd` Das Protokoll zeigt möglicherweise die Fehlermeldung an `error: cannot create url_parser context for http_parser/libhttp_parser`. Dies ist ein bekanntes Slurm-Problem, das auch dann auftritt, wenn `CommunicationParameters=disable_http` ist gesetzt. Der Fehler kann getrost ignoriert werden und hat keinen Einfluss auf den Clusterbetrieb.

Weitere Informationen zu Slurm 25.11 finden Sie in den folgenden Publikationen:

- Ankündigung der Veröffentlichung von SchedMD: https://www.schedmd.com/slurm-version-25-11-0-is-now-available/
- Versionshinweise zu SchedMD: https://github.com/SchedMD/slurm/blob/slurm-25.11/RELEASE_NOTES.md

Slurm 25.05

Änderungen implementiert in AWS STCK

- Der Slurm requeue_on_resume_failure SchedulerParameter ist jetzt standardmäßig aktiviert.
- „stderr“ wurde als Option für entfernt LogTimeFormat, da es in Slurm 25.05 deaktiviert war.
- AWS PCS unterstützt Multi-cluster Sackd-Konfiguration: Der Anmeldeknoten kann auf mehrere Cluster zugreifen.

Weitere Informationen zu Slurm 25.05 finden Sie in den folgenden Veröffentlichungen:

- Ankündigung der Veröffentlichung von SchedMD: https://www.schedmd.com/slurm-version-25-05-0-is-now-available/
- Versionshinweise zu SchedMD: https://github.com/SchedMD/slurm/blob/slurm-25-05-0-1/RELEASE_NOTES.md

Slurm 24.11

Änderungen implementiert in AWS STCK

- AWS PCS unterstützt die Buchhaltung von Slurm. Weitere Informationen finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.

Weitere Informationen zu Slurm 24.11 finden Sie in den folgenden Publikationen:

- Ankündigung der Veröffentlichung von SchedMD
- Versionshinweise zu SchedMD

Slurm 24.05

Änderungen implementiert in AWS STCK

- Das neue Slurm Step Manager-Modul ist jetzt standardmäßig aktiviert in AWS 5 STCK. Dieses Modul bietet erhebliche Vorteile, da die Schrittverwaltung von der zentralen Steuerung auf die Rechenknoten verlagert wird, wodurch die Parallelität des Systems in Umgebungen mit starker Step-Nutzung erheblich verbessert wird. Um diese Konfiguration zu unterstützen und besser zu

isolieren `Prolog` und `Epilog` Prozessausführung, neue Prolog-Flags (`Contain`, `Alloc`) sind aktiviert.

- Die hierarchische Kommunikation vom Controller zu den Rechenknoten ist aktiviert, um die knoteninterne Slurm-Kommunikation zu optimieren, was die Skalierbarkeit und Leistung verbessert. Darüber hinaus verwendet die Routing-Konfiguration jetzt Partitionsknotenlisten für die Kommunikation vom Controller anstelle des Standard-Routing-Algorithmus des Plugins, wodurch die Systemstabilität verbessert wird.
- Ein neues Hash-Plugin `HashPlugin=hash/sha3` ersetzt das vorherige `hash/k12 plugin`. Dies ist jetzt standardmäßig aktiviert in AWS PCS-Cluster.
- Die Slurm-Controller-Protokolle enthalten jetzt erweiterte Überwachungsfunktionen für alle eingehenden Remote Procedure Calls (RPC) an `slurmctld`. Die Protokolle enthalten die Quelladresse, den authentifizierten Benutzer und den RPC-Typ vor der Verbindungsverarbeitung.

Weitere Informationen zu Slurm 24.05 finden Sie in den folgenden Publikationen:

- Ankündigung der Veröffentlichung von SchedMD
- Versionshinweise zu SchedMD

Slurm 23.11

Slurm-Einstellungen kannst du ändern in AWS 5 STCK

- Das `SuspendTime` standardmäßig `60`. Benutze das AWS STCK `scaleDownIdleTimeInSeconds` Konfigurationsparameter, um es einzustellen. Weitere Informationen finden Sie in der `scaleDownIdleTimeInSeconds` Parameter des `ClusterSlurmConfiguration` Datentyp in der AWS PCS-API-Referenz.
- Das `MaxJobCount` und `MaxArraySize` basiert auf der Größe, die Sie für den Cluster wählen. Weitere Informationen finden Sie in der `size` Parameter des `CreateCluster` API-Aktion in der AWS PCS-API-Referenz.
- Das `SelectTypeParameters` Die Slurm-Einstellung ist standardmäßig `CR_CPU`. Sie können es als Wert angeben für `slurmCustomSettings` um es festzulegen, wenn Sie einen Cluster erstellen. Weitere Informationen finden Sie in der `slurmCustomSettings` Parameter des `CreateCluster` API-Aktion und SlurmCustomSetting in der AWS PCS-API-Referenz.
- Du kannst einstellen `Prolog` und `Epilog` auf Cluster-Ebene. Sie können es als Wert angeben für `slurmCustomSettings` um es festzulegen, wenn Sie einen Cluster erstellen. Weitere

Informationen finden Sie unter `CreateCluster` und SlurmCustomSetting in der AWS PCS-API-Referenz.

- Du kannst einstellen `Weight` und `RealMemory` auf Gruppenebene des Rechenknotens. Sie können es als Wert angeben für `slurmCustomSettings` um ihn festzulegen, wenn Sie eine Compute-Knotengruppe erstellen. Weitere Informationen finden Sie unter `CreateComputeNodeGroup` und SlurmCustomSetting in der AWS PCS-API-Referenz.

Häufig gestellte Fragen zu Slurm-Versionen in AWS 5 STCK

AWS PCS unterstützt weiterhin mehrere Slurm-Versionen. Wenn eine neue Slurm-Version eingeführt wird, AWS PCS bietet technischen Support und Sicherheitspatches, bis diese Version das Ende des Supports (EOS) von SchedMD erreicht. AWS PCS bezeichnet das EOS-Datum für eine Slurm-Version als End of Life (EOL), um konsistent zu sein AWS Terminologie.

Wie lang dauert AWS PCS unterstützt eine Slurm-Version?

AWS Die PCS-Unterstützung für Slurm-Versionen entspricht den Support-Zyklen von SchedMD für Hauptversionen. AWS PCS unterstützt die aktuelle Version und die beiden neuesten vorherigen Hauptversionen. Wenn SchedMD eine neue Hauptversion veröffentlicht, AWS PCS beendet die Unterstützung für die älteste unterstützte Version. AWS PCS veröffentlicht neue Hauptversionen von Slurm so schnell wie möglich, aber es kann zu Verzögerungen zwischen der Veröffentlichung von SchedMD und seiner Verfügbarkeit in kommen AWS STCK.

Wie erhalten meine Cluster neue Slurm-Patch-Versionen?

Um Fehler und Sicherheitskorrekturen zu beheben, AWS PCS ist so konzipiert, dass es automatisch Patches auf Cluster-Controller anwendet, die unter internen, diensteigenen Konten ausgeführt werden. So installieren Sie Patches auf EC2-Instances in Ihrem AWS-Konto, aktualisieren Sie das Amazon Machine Image (AMI) für Ihre Rechenknotengruppen und aktualisieren Sie die Rechenknotengruppen, um das aktualisierte AMI zu verwenden. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

> **Note**
>
> Slurm-Controller sind nicht verfügbar, solange wir sie aktualisieren. Laufende Jobs sind nicht betroffen. Jobs, die eingereicht wurden, bevor der Controller des Clusters nicht mehr verfügbar war, werden zurückgehalten, bis der Controller verfügbar ist.

Wie werde ich über ein bevorstehendes EOL-Event in der Slurm-Version informiert?

Wir senden Ihnen 6 Monate vor dem EOL-Datum eine E-Mail-Nachricht. Wir senden Ihnen jeden Monat vor dem EOL-Datum eine E-Mail-Nachricht, mit einer letzten E-Mail-Nachricht 1 Woche vor dem EOL-Datum. Nach dem EOL-Datum senden wir 12 Monate lang monatliche E-Mail-Nachrichten an laufende Kunden AWS PCS-Cluster mit EOL Slurm-Versionen. Wir könnten einen Cluster mit einer EOL-Slurm-Version aussetzen, wenn Sicherheitslücken für diese Version identifiziert werden.

Wie kann ich feststellen, ob auf der von meinem Cluster verwendeten Slurm-Version eine EOL-Slurm-Version ausgeführt wird?

Wir senden Ihnen eine E-Mail-Nachricht, um Sie darüber zu informieren, dass Sie einen laufenden Cluster mit einer EOL-Slurm-Version haben. Wir senden eine Warnung an die AWS Health Dashboard Benachrichtigungen, die die Details Ihrer Cluster mit EOL Slurm-Versionen enthalten. Sie können auch die verwenden AWS PCS-Konsole zur Identifizierung der Cluster mit EOL Slurm-Versionen.

Was muss ich tun, wenn meine Slurm-Version fast oder älter als EOL ist?

Erstellen Sie einen neuen Cluster mit einer neueren unterstützten Version von Slurm und aktualisieren Sie die Slurm-Version in den AMIs Ihrer Compute-Knotengruppe. Die Slurm-Version in Ihren AMIs und den laufenden EC2-Instances darf nicht mehr als 2 Versionen hinter der Slurm-Version des Clusters liegen. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Was passiert, wenn ich bis zum EOL-Datum nicht auf eine neuere Version von Slurm umsteige?

Mit einer EOL Slurm-Version kannst du keine neuen Cluster erstellen. Bestehende Cluster können bis zu 12 Monate ohne AWS Unterstützung, und es sind keine sofortigen Maßnahmen erforderlich, um ihren Betrieb aufrechtzuerhalten. Nach dem EOL-Datum können Support, Sicherheitsupdates und Verfügbarkeit nicht garantiert werden. Wir könnten einen Cluster aus Sicherheitsgründen aussetzen. Wir empfehlen Ihnen dringend, eine unterstützte Slurm-Version zu verwenden, um die Sicherheit und den Support für Ihre AWS PCS-Cluster.

Was sind die Risiken des Betriebs eines Clusters mit EOL Slurm-Versionen?

Cluster mit EOL Slurm-Versionen bergen erhebliche Sicherheits- und Betriebsrisiken. Ohne die aktive Überwachung von SchedMD bleiben Sicherheitslücken möglicherweise unentdeckt oder werden nicht behoben. Wenn kritische Sicherheitslücken entdeckt werden, können wir Ihre Cluster sofort sperren.

Was passiert mit meinen Jobs, Cluster-Rechen-, Speicher- und Netzwerkressourcen, wenn mein Cluster suspendiert wird?

Alle Ressourcen werden verwaltet von AWS PCS sind beendet. Dazu gehören der Slurm-Controller, Compute Node Groups und EC2-Instances. Alle Jobs, die auf Recheninstanzen ausgeführt werden, werden sofort beendet, und der Cluster wechselt in einen angehaltenen Zustand. Customer-managed Ressourcen, wie z. B. externe Dateisysteme, bleiben intakt. Sie können das verwenden AWS PCS-Konsole und API-Aktionen für den Zugriff auf die Cluster-Konfiguration.

Kann ich einen unterbrochenen Cluster neu starten, um die verbleibenden Jobs fortzusetzen?

Nein, Sie können einen unterbrochenen Cluster nicht neu starten. Sie können die Konfiguration Ihres suspendierten Clusters verwenden, um einen neuen Cluster mit einer unterstützten Slurm-Version zu erstellen. Sie können Ihre verbleibenden Jobs ausführen, wenn Sie sie in einem externen Dateisystem gespeichert haben.

Kann ich eine Verlängerung über die 12-monatige Nachfrist hinaus beantragen?

Nein, Sie können keine Verlängerung beantragen, um Ihren Cluster über die 12-monatige Kulanzfrist hinaus laufen zu lassen. Wir stellen die verlängerte Zeit zur Verfügung, um Ihnen bei der Umstellung auf eine unterstützte Slurm-Version zu helfen. Um Unterbrechungen Ihres Clusterbetriebs zu vermeiden, empfehlen wir Ihnen, zu wechseln, bevor Ihre Slurm-Version EOL erreicht.

Slurm-Konfiguration in AWS STCK

AWS PCS verwaltet die Slurm-Konfiguration Ihres Clusters. Der Cluster-Controller enthält die autoritativen Konfigurationsdateien, darunter `slurm.conf`, `cgroup.conf`, und `slurmdbd.conf`. AWS PCS generiert diese Dateien aus Ihren Cluster-, Warteschlangen- und Rechenknotengruppeneinstellungen und aktualisiert sie, wenn sich Ihr Cluster ändert.

Modus ohne Konfiguration

AWS PCS führt Slurm im konfigurationslosen Modus aus. Im Modus ohne Konfiguration rufen Knoten die Konfiguration direkt vom Cluster-Controller ab (`slurmctld`). Weitere Informationen finden Sie unter Slurm ohne Konfiguration auf der SchedMD-Website.

- Auf Rechenknoten `slurmd` ruft die Konfiguration vom Controller ab, wenn er gestartet wird.
- Auf Anmeldeknoten der Slurm-Daemon für Authentifizierung und Anmeldeinformationen (`sackd`) ruft die Konfiguration ab, sodass Client-Befehle wie `sbatch`, `srun`, und `scontrol` kann es benutzen.

Slurm speichert die Dateien, die es auf dem Knoten abruft, im Cache und erstellt einen symbolischen Link zu ihnen unter `/run/slurm/conf`. Die Konfiguration, mit der ein Knoten läuft, ist `/run/slurm/conf/slurm.conf`.

Um die Konfiguration zu überprüfen, mit der ein Knoten ausgeführt wird, verwenden Sie den folgenden Befehl auf dem Knoten.

```
scontrol show config
```

Mit AWS PCS, Sie können eine begrenzte Gruppe von Slurm-Einstellungen konfigurieren. Die Einstellungen, die jede Konfigurationsdatei unterstützt, finden Sie in den folgenden Themen.

Themen

- Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK
- Konfiguration benutzerdefinierter Cgroup-Einstellungen in AWS STCK
- Konfiguration benutzerdefinierter SlurmDBD-Einstellungen in AWS 5 STÜCK
- Konfiguration benutzerdefinierter GRES-Einstellungen in AWS 5 STCK

Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK

Verwenden Sie benutzerdefinierte Slurm-Einstellungen, um zusätzliche Slurm-Parameter für Cluster-, Queue- und Compute Node Group-Ressourcen zu konfigurieren. Diese Version bietet Unterstützung für Slurm-Einstellungen für Warteschlangenressourcen und bietet so eine granulare Kontrolle über partitionsspezifisches Verhalten.

Vorteile benutzerdefinierter Slurm-Einstellungen

Benutzerdefinierte Slurm-Einstellungen bieten eine ausgefeilte Kontrolle über Ihre AWS PCS-based HPC-Umgebung. Sie können eine detaillierte Abrechnung implementieren, Zugriffskontrollen durchsetzen und die Workload-Ausführung mithilfe von QoS-Konfigurationen und Präventionsrichtlinien optimieren. Diese Funktionen stellen sicher, dass wichtige Aufgaben die notwendigen Ressourcen erhalten und gleichzeitig eine effiziente Cluster-Auslastung gewährleistet wird. Ob Sie es schaffen GPU-accelerated Workloads, die Implementierung von Fair-Share-Scheduling oder die Kontrolle von Job-Lifecycles — benutzerdefinierte Einstellungen helfen Ihnen dabei, Ihre HPC-Infrastruktur auf betriebliche Anforderungen und Forschungsziele abzustimmen.

Konfiguration benutzerdefinierter Einstellungen

Benutzerdefinierte Slurm-Einstellungen können über das konfiguriert werden AWS Konsole, CLI oder SDKs während der Ressourcenerstellung oder später durch Aktualisierungsvorgänge geändert.

AWS-Managementkonsole

Navigieren Sie zu Zusätzliche Scheduler-Einstellungen auf der Erstellungs- oder Bearbeitungsseite für einen beliebigen Ressourcentyp (Cluster, Warteschlange oder Compute-Knotengruppe).

Um eine neue Einstellung hinzuzufügen

1. Wählen Sie Neue Einstellung hinzufügen.
2. Wählen Sie ein Parameter Name aus dem Drop-down-Menü (das kurze Parameterbeschreibungen enthält).
3. Geben Sie den entsprechenden Wert ein.

Um eine benutzerdefinierte Einstellung rückgängig zu machen

1. Wählen Sie Entfernen neben dem entsprechenden parameter/value Paar.
2. Erstellen oder aktualisieren Sie die Ressource.

AWS CLI

Verwenden Sie für die programmatische Verwaltung benutzerdefinierter Einstellungen das `SlurmCustomSettings` Feld in Erstellungs- oder Aktualisierungsvorgängen.

Example— Aktualisierung der Prolog Parameter auf einem Cluster

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'SlurmCustomSettings=[{parameterName=Prolog,parameterValue="/path/to/prolog.sh"}]'
```

Example— Einstellung einer Warteschlange als Standard auf einem Cluster

```
aws pcs update-queue \
    --cluster-identifier my-cluster \
```

```
    --queue-identifier my-queue \
    --slurm-configuration
 'SlurmCustomSettings=[{parameterName=Default,parameterValue=YES}]'
```

Example— Einstellung benutzerdefiniert Funktionen auf einer Compute-Knotengruppe

```
aws pcs update-compute-node-group \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-cng-1 \
    --slurm-configuration \
    'SlurmCustomSettings=[{parameterName=Features,parameterValue="gpu,nvme"}]'
```

Validierung und Fehlerbehandlung

AWS PCS implementiert einen mehrstufigen Validierungsprozess für benutzerdefinierte Slurm-Einstellungen. Sowohl beim Erstellen als auch beim Aktualisieren führen wir synchrone Validierungen durch, die Folgendes beinhalten:

- Field-level Prüfungen: Wir überprüfen einzelne Einstellungen auf korrekte Datentypen, zulässige Werte und Formatanforderungen. Zum Beispiel stellen wir sicher, dass Zeitwerte das richtige Slurm-Format haben und boolesche Werte akzeptierte boolesche Slurm-Repräsentationen verwenden.
- Context-aware Validierungen: Einige Einstellungen werden mit dem breiteren Konfigurationskontext verglichen. Zum Beispiel sind bestimmte Parameter nur gültig, wenn die Slurm-Buchhaltung aktiviert ist.
- Inter-setting Konsistenz: Wir stellen sicher, dass sich gegenseitig ausschließende Optionen nicht zusammen gesetzt sind und dass die voneinander abhängigen Einstellungen korrekt konfiguriert sind.

Wenn die Validierung fehlschlägt, erhalten Sie eine `ValidationException` mit einem bestimmten Fehlercode (z. B. InvalidInput), eine klare Fehlermeldung, die das Problem beschreibt, und eine Liste der ungültigen Felder und ihrer jeweiligen Fehlerdetails.

Während dieser ersten Überprüfung werden zwar viele Probleme erkannt, aber einige komplexe Interaktionen zwischen den Einstellungen werden möglicherweise erst bei der Anwendung der Konfiguration sichtbar. In solchen Fällen schlägt der Vorgang mit einer informativen Fehlermeldung fehl, und alle teilweisen Änderungen werden rückgängig gemacht.

Einschränkungen

AWS PCS verwendet zum Schutz der Servicesicherheit und Betriebsstabilität einen Zulassungsansatz. Einstellungen, die die Sicherheit des Dienstkontos gefährden oder die Funktionen der verwalteten Dienste beeinträchtigen könnten, sind eingeschränkt. Wir evaluieren jedoch kontinuierlich die Kundenbedürfnisse und können auf der Grundlage von Kundenfeedback Unterstützung für zusätzliche Einstellungen hinzufügen.

Themen

- Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster
- Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen
- Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen
- Problembehandlung bei benutzerdefinierten Slurm-Einstellungen in AWS 5 STCK

Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster

Die folgenden benutzerdefinierten Slurm-Einstellungen werden auf Cluster-Ebene unterstützt:

- AccountingStorageEnforce
- AccountingStorageTRES
- AccountingStoreFlags
- AuthAltParameters
- CliFilterParameters

> **Note**
>
> Weitere Informationen zu CLI-Filtern finden Sie unter AWS PCS, siehe Konfigurieren Sie die Slurm CLI Filter Plugins auf einem AWS PCS-Cluster.

- CliFilterPlugins

> **Note**
>
> Weitere Informationen zu CLI-Filtern finden Sie unter AWS PCS, siehe Konfigurieren Sie die Slurm CLI Filter Plugins auf einem AWS PCS-Cluster.

- CommunicationParameters

> **Important**
>
> AWS PCS deaktiviert standardmäßig den HTTP-Endpunkt. Um ihn zu aktivieren, geben Sie an `enable_http`.

- DefMemPerCPU
- Epilog
- EnforcePartLimits
- FairShareDampeningFactor
- FirstJobId
- HealthCheckInterval
- HealthCheckNodeState

> **Note**
>
> AWS PCS unterstützt die `REBOOT_ONLY` Wert, der den ausführt `HealthCheckProgram` erst nach einem Node-Neustart, auf Slurm-Version 26.05 und höher. `REBOOT_ONLY` schließt sich mit jedem anderen Wert gegenseitig aus.

- HealthCheckProgram
- JobRequeue
- LaunchParameters
- Licenses
- MetricsType

> **Note**
>
> Weitere Informationen zu Metriken finden Sie unter AWS PCS, siehe Slurm Metriken in AWS STCK.

- MinJobAge

> **Note**
>
> AWS PCS unterstützt einen Mindestwert von 5 Sekunden für `MinJobAge`.

- OverTimeLimit
- PreemptExemptTime
- PreemptMode
- PreemptParameters
- PreemptType
- PriorityCalcPeriod
- PriorityDecayHalfLife
- PriorityFavorSmall
- PriorityFlags
- PriorityMaxAge
- PriorityUsageResetPeriod
- PriorityWeightAge
- PriorityWeightAssoc
- PriorityWeightFairshare
- PriorityWeightJobSize
- PriorityWeightPartition
- PriorityWeightQOS
- PriorityWeightTRES
- PrivateData
- Prolog
- PrologFlags

> **Note**
>
> Zu verwenden X11 Weiterleitung (das heißt, wenn Sie das aktivieren X11 Flagge an PrologFlags), muss das xauth-Hilfsprogramm auf dem Compute-Knotengruppen-AMI (dem Worker-AMI) installiert sein. Ohne es X11 Die Weiterleitung funktioniert nicht. Auf Amazon Linux wird xauth bereitgestellt von `xorg-x11-xauth` Paket. Um es zu installieren, verwenden Sie zum Beispiel den Systempaketmanager `sudo yum install xorg-x11-xauth`.

- RequeueExit
- RequeueExitHold

- SchedulerParameters
- SelectTypeParameters

> **Note**
>
> AWS PCS unterstützt die `CR_Socket` und `CR_Socket_Memory` Werte auf Slurm-Version 25.11 und höher.

- SlurmctldParameters

> **Note**
>
> AWS PCS unterstützt die `enable_stepmgr` Wert. Es erfordert `PrologFlags` nicht angegeben oder einzubeziehen `Contain`.

- SrunPortRange
- TaskEpilog
- TaskPluginParam
- TaskProlog
- TrackWCKey
- UnkillableStepProgram
- UnkillableStepTimeout

Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen

Die folgenden benutzerdefinierten Slurm-Einstellungen werden auf der Gruppenebene des Rechenknotens unterstützt:

- CpuSpecList
- Features
- Gres

> **Note**
>
> Verwenden `Gres` um das GRES zu deklarieren, das die Compute-Knotengruppe für planbar macht. Jede Ressource, die Sie hier benennen, muss auch als deklariert werden

`gres.conf` aufzeichnen in `gresCustomSettings`, und die Anzahl und Typen der beiden müssen konsistent sein. Informationen zur Konfiguration der zugrunde liegenden GRES-Geräte und eine vollständige Liste der Einschränkungen finden Sie unter Konfiguration benutzerdefinierter GRES-Einstellungen in AWS 5 STCK.

- MemSpecLimit
- Parameters

> **Note**
>
> AWS PCS unterstützt `Parameters` auf Slurm Version 25.11 und höher.

- RealMemory
- Sockets

> **Note**
>
> AWS PCS unterstützt `Sockets` auf Slurm Version 25.11 und höher.

- Weight

Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen

Die folgenden benutzerdefinierten Slurm-Einstellungen werden auf Warteschlangenebene unterstützt:

> **Important**
>
> QOS-Werte, auf die verwiesen wird in `AllowQos`, `DenyQos`, oder `QOS` Einstellungen müssen bereits in der Slurm-Buchhaltungsdatenbank vorhanden sein. Andernfalls kann Slurm die Konfiguration möglicherweise nicht anwenden. Weitere Informationen zur Slurm-Buchhaltung finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.

- AllowAccounts
- AllowQos
- Default
- DefaultTime

- DenyAccounts
- DenyQos
- Exclusive

> **Note**
>
> `Exclusive` ersetzt `ExclusiveUser` und `OverSubscribe=EXCLUSIVE`. `Exclusive=NODE` und `Exclusive=TOPO` implizieren `OverSubscribe=NO`.

- ExclusiveUser

> **Note**
>
> Slurm ist abwertend `ExclusiveUser` ab Version 26.05 zugunsten von `Exclusive`. AWS PCS unterstützt es weiterhin, aber wir empfehlen Ihnen, zu migrieren `Exclusive`.

- GraceTime
- MaxTime
- OverSubscribe
- OverTimeLimit
- PowerDownOnIdle
- PreemptMode
- PriorityJobFactor
- PriorityTier
- QOS
- TRESBillingWeights

Problembehandlung bei benutzerdefinierten Slurm-Einstellungen in AWS 5 STCK

Wenn beim Erstellen oder Aktualisieren Fehler auftreten AWS PCS-Ressourcen mit benutzerdefinierten Slurm-Einstellungen. Sie können die Protokollierung verwenden, um die Probleme zu diagnostizieren und zu lösen.

Behebung inkompatibler benutzerdefinierter Slurm-Einstellungen

Problem: Sie erhalten eine Fehlermeldung ähnlich der folgenden, wenn Sie Cluster-, Compute-Knotengruppen- oder Warteschlangenoperationen ausführen:

```
{OPERATION} failed. The Slurm custom settings of the cluster might be incompatible.
 Check the settings and try again.
```

Dieser Fehler kann bei den folgenden Vorgängen auftreten:

- CreateCluster
- CreateComputeNodeGroup
- UpdateComputeNodeGroup
- CreateQueue
- UpdateQueue

Lösung: Aktivieren Sie die Protokollierung, um das spezifische Problem zu verstehen und die inkompatiblen Einstellungen zu beheben.

Um inkompatible benutzerdefinierte Slurm-Einstellungen zu beheben

1. Erstellen Sie den Cluster, falls er noch nicht existiert, oder stellen Sie sicher, dass sich Ihr vorhandener Cluster in einem Zustand befindet, in dem die Protokollierung aktiviert werden kann.
2. Aktivieren Sie die Protokollierung für Ihren Cluster. Detaillierte Anweisungen finden Sie unter Protokollierung und Überwachung für AWS 5 STCK.

   > **Note**
   >
   > Die Protokollierung kann aktiviert werden, sobald der Cluster erstellt wird.

3. Überprüfen Sie die Protokolle, um das spezifische Slurm-Konfigurationsproblem zu identifizieren, das die Inkompatibilität verursacht hat.
4. Korrigieren Sie die inkompatiblen benutzerdefinierten Einstellungen anhand der Protokollinformationen und wiederholen Sie den Vorgang.

Informationen zu den unterstützten benutzerdefinierten Slurm-Einstellungen finden Sie unter:

- Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Cluster
- Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen
- Benutzerdefinierte Slurm-Einstellungen für AWS PCS-Warteschlangen

Konfiguration benutzerdefinierter Cgroup-Einstellungen in AWS STCK

Slurm verwendet das Linux-Subsystem cgroup, um Ressourcen für Jobs wie Arbeitsspeicher, CPU-Kerne, Geräte und Swap-Space zu verwalten und einzuschränken. AWS Mit PCS können Sie anpassen `cgroup.conf` Einstellungen auf Cluster-Ebene über den `CgroupCustomSettings` Eigentum von `SlurmConfiguration` während der Clustererstellung oder -aktualisierung.

Konfiguration der cgroup-Einstellungen

Benutzerdefinierte Cgroup-Einstellungen können über das konfiguriert werden AWS Konsole, CLI oder SDKs während der Clustererstellung oder später durch Aktualisierungsvorgänge geändert.

AWS-Managementkonsole

Navigieren Sie zu Zusätzliche Scheduler-Einstellungen auf der Seite zum Erstellen oder Bearbeiten einer Cluster-Ressource.

Um eine neue Einstellung hinzuzufügen

1. Wählen Sie Neue Einstellung hinzufügen.
2. Wählen Sie ein Parameter Name aus dem Drop-down-Menü (das kurze Parameterbeschreibungen enthält).
3. Geben Sie den entsprechenden Wert ein.

Um eine benutzerdefinierte Einstellung rückgängig zu machen

1. Wählen Sie Entfernen neben dem entsprechenden parameter/value Paar.
2. Erstellen oder aktualisieren Sie die Ressource.

AWS CLI

Für die programmatische Verwaltung der Cgroup-Einstellungen verwenden Sie `CgroupCustomSettings` Feld in den Vorgängen zum Erstellen oder Aktualisieren von Clustern.

Example— Einstellung RAM-Speicherplatz einschränken auf einem Cluster

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'CgroupCustomSettings=[{parameterName=ConstrainRAMSpace,parameterValue="yes"}]'
```

Unterstützte Cgroup-Einstellungen für Cluster

Die folgenden benutzerdefinierten Cgroup-Einstellungen werden auf Cluster-Ebene unterstützt:

- AllowedRAMSpace
- AllowedSwapSpace
- ConstrainCores
- ConstrainDevices
- ConstrainRAMSpace
- ConstrainSwapSpace
- IgnoreSystemd
- MaxRAMPercent
- MaxSwapPercent
- MinRAMSpace
- SignalChildrenProcesses

Konfiguration benutzerdefinierter SlurmDBD-Einstellungen in AWS 5 STÜCK

Der Datenbank-Daemon (slurmdbd) von Slurm verwaltet Buchhaltungsdaten, Datenaufbewahrungsrichtlinien und Datenschutzkontrollen. AWS Mit PCS können Sie anpassen `slurmdbd.conf` Einstellungen auf Cluster-Ebene über den `SlurmdbdCustomSettings` Eigentum von `SlurmConfiguration` während der Clustererstellung oder -aktualisierung.

Konfiguration der slurmdbd-Einstellungen

Die benutzerdefinierten Einstellungen von Slurmdbd können über das konfiguriert werden AWS Konsole, CLI oder SDKs während der Clustererstellung oder später durch Aktualisierungsvorgänge geändert.

AWS-Managementkonsole

Navigieren Sie zu Zusätzliche Scheduler-Einstellungen auf der Seite zum Erstellen oder Bearbeiten einer Cluster-Ressource.

Um eine neue Einstellung hinzuzufügen

1. Wählen Sie Neue Einstellung hinzufügen.
2. Wählen Sie ein Parameter Name aus dem Drop-down-Menü (das kurze Parameterbeschreibungen enthält).
3. Geben Sie den entsprechenden Wert ein.

Um eine benutzerdefinierte Einstellung rückgängig zu machen

1. Wählen Sie Entfernen neben dem entsprechenden parameter/value Paar.
2. Erstellen oder aktualisieren Sie die Ressource.

AWS CLI

Für die programmatische Verwaltung der slurmdbd-Einstellungen verwenden Sie den `SlurmdbdCustomSettings` Feld in Cluster-Vorgängen zum Erstellen oder Aktualisieren.

Example— Einstellung Track-WCKEY auf einem Cluster

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'SlurmdbdCustomSettings=[{parameterName=TrackWCKey,parameterValue="yes"}]'
```

Unterstützte slurmdbd-Einstellungen für Cluster

Die folgenden benutzerdefinierten Slurmdbd-Einstellungen werden auf Clusterebene unterstützt:

- AllowNoDefAcct
- AllResourcesAbsolute
- CommitDelay
- DefaultQOS
- MaxQueryTimeRange

- Parameters
- PrivateData
- PurgeEventAfter
- PurgeJobAfter
- PurgeJobEnvAfter
- PurgeJobScriptAfter
- PurgeResvAfter
- PurgeStepAfter
- PurgeSuspendAfter
- PurgeTXNAfter
- PurgeUsageAfter
- TrackWCKey

Konfiguration benutzerdefinierter GRES-Einstellungen in AWS 5 STCK

Slurm verwendet `gres.conf` um generische Ressourcen (GRES) wie GPUs zu verwalten, Multi-Process Service (MPS) und Multi-Instance GPU-Geräte (MIG) auf Rechenknoten. AWS Mit PCS können Sie Folgendes anpassen `gres.conf` durch die `gresCustomSettings` Eigentum von `slurmConfiguration` während der Erstellung oder Aktualisierung der Compute-Knotengruppe.

Im Gegensatz zu anderen benutzerdefinierten Einstellungen, bei denen es sich um eine einfache Liste von Parameter- und Wertepaaren handelt, `gresCustomSettings` ist eine Liste von Aufzeichnungen. Jeder Datensatz ist ein Satz von `gres.conf` Attribute, die ein GRES-Gerät oder eine GRES-Ressource beschreiben. Zum Beispiel der Datensatz `Name=gpu Type=a100 File=/dev/nvidia[0-7]` deklariert acht A100-GPUs.

In Slurma, ein `gres.conf` Der Datensatz gilt für jeden Knoten im Cluster, es sei denn, er gibt einen an `NodeName=` Präfix. AWS PCS fügt immer das hinzu `NodeName=` Präfix für die Rechenknotengruppe, Sie geben es also nicht an, und jeder Datensatz, den Sie angeben, gilt nur für diese Rechenknotengruppe. AWS PCS unterstützt keine clusterweite `gres.conf` Aufzeichnungen.

Konfiguration der gres.conf-Einstellungen

`gres.conf` benutzerdefinierte Einstellungen können über das konfiguriert werden AWS Konsole, CLI oder SDKs während der Erstellung der Compute-Knotengruppe oder später durch Aktualisierungsvorgänge geändert.

AWS-Managementkonsole

Navigieren Sie zu Zusätzliche Scheduler-Einstellungen auf der Seite „Erstellen oder Bearbeiten“ für eine Compute-Knotengruppen-Ressource. Geben Sie eins ein `gres.conf` Datensatz pro Zeile, nativ `gres.conf` Formular (zum Beispiel `Name=gpu Type=a100 File=/dev/nvidia[0-7]`).

AWS CLI

Für die programmatische Verwaltung von `gres.conf` Einstellungen, verwenden Sie die `gresCustomSettings` Feld in Operationen zum Erstellen oder Aktualisieren von Rechenknotengruppen. Jedes Element der Liste ist eins `gres.conf` aufzeichnen.

Example— Deklarieren von GPU-Geräten in einer Rechenknotengruppe

```
aws pcs update-compute-node-group \
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-cng-1 \
    --slurm-configuration \
    'gresCustomSettings=[{Name=gpu,Type=a100,File=/dev/nvidia[0-7]}]'
```

Unterstützte gres.conf-Einstellungen für Compute-Knotengruppen

Folgendes `gres.conf` Attribute werden in jedem Datensatz auf Gruppenebene des Rechenknotens unterstützt:

- AutoDetect

  > **Note**
  >
  > AWS PCS unterstützt `AutoDetect` auf Slurm Version 25.11 und höher.

- Name
- Type
- Count
- Cores
- Links
- Flags
- File

- MultipleFiles

Die Bedeutung und die akzeptierten Werte der einzelnen Attribute finden Sie im Slurm gres.conf Dokumentation. AWS PCS validiert Ihre Aufzeichnungen anhand derselben Regeln, die Slurm anwendet. Insbesondere muss jeder Datensatz entweder einen deklarieren `Name` oder ein aktives `AutoDetect` Wert und ein GPU-Datensatz (`Name=gpu`) muss Folgendes angeben `File` oder `MultipleFiles` es sei denn, ein aktiver `AutoDetect` Der Wert ermittelt die Geräte.

Ein aktiv `AutoDetect` Wert ist ein beliebiger Wert, der die Erkennung aktiviert, z. B. `nvml` oder `full`. `AutoDetect=off` schaltet die Erkennung aus, sodass es sich nicht um einen aktiven Wert handelt und in einem Datensatz nicht eigenständig verwendet werden kann.

Wie AWS PCS wendet Ihre gres.conf-Datensätze an

Eine GRES-Konfiguration erscheint an zwei Stellen in der Slurm-Konfiguration und AWS PCS schreibt beide:

- Die `gres.conf` Aufzeichnungen, die Sie einreichen `gresCustomSettings` beschreiben Sie die Geräte, die auf dem Knoten vorhanden sind.
- Die `Gres` Einsetzen `slurm.conf` deklariert, was der Scheduler zuweisen kann, in der folgenden Form `name[:type]:count`. Sie geben es als benutzerdefinierte Slurm-Einstellung für die Compute-Knotengruppe an, in `slurmCustomSettings`. Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.

AWS PCS rendert Ihre Aufzeichnungen in `gres.conf` und gleicht sie mit der GRES-Konfiguration ab, die es verwaltet:

- Jeder GPU-Datensatz, den Sie deklarieren, ersetzt den Standard-GPU-Datensatz AWS PCS generiert für die Compute-Knotengruppe. Ein GPU-Datensatz ist ein Datensatz, der Folgendes angibt `Name=gpu` oder ein `AutoDetect` Attribut.
- Datensätze für andere Ressourcen, wie `mps`, `shard`, oder ein eigener Ressourcenname, werden neben dem hinzugefügt AWS PCS-Standardwerte.
- AWS PCS leitet den ab `GresTypes` Einstellung und die `gres/*` Einträge des `AccountingStorageTRES` Einsetzen `slurm.conf` aus dem `Gres` Einstellung.

In Slurm-Version 26.05 und höher AWS PCS-Sätze `AutoDetect=full` standardmäßig auf GPU-Compute-Knotengruppen, wenn das Socket-Layout des Knotens bekannt ist, entweder weil Sie

`Sockets` in `slurmCustomSettings` für die Compute-Knotengruppe, oder weil AWS PCS hat die CPU-Topologie automatisch erkannt. Wenn das Socket-Layout nicht verfügbar ist, AWS PCS gibt seinen GPU-Datensatz ohne `AutoDetect` stattdessen.

Einschränkungen zwischen gres.conf und der Gres-Einstellung

Slurm benötigt beide Seiten der GRES-Konfiguration, um eine Ressource zu planen, und eine Ressource, die nur in einer von ihnen vorkommt, funktioniert nicht. Ein Eintrag in der `Gres` Einsetzen `slurm.conf` ohne `gres.conf` Ein Datensatz dahinter sorgt dafür, dass der Knoten weniger Ressourcen meldet als konfiguriert sind, und der Knoten wird leer. Ein Datensatz ohne `Gres` Nach einem Eintrag ist das Gerät zwar auf dem Knoten vorhanden, kann aber nicht geplant werden.

AWS PCS validiert die beiden miteinander, wenn Sie eine Rechenknotengruppe erstellen oder aktualisieren, und lehnt die Anfrage in den folgenden Fällen ab:

- A `slurm.conf` `Gres` Ein Eintrag benennt eine Ressource, die nein `gres.conf` Datensatz deklariert. Das beinhaltet `gpu`: ein `gpu` Der Eintrag erfordert einen GPU-Datensatz in `gresCustomSettings`, auch wenn die Anzahl `0`.
- EIN `slurm.conf` `Gres` Ein Eintrag deklariert eine höhere Anzahl einer Ressource als `gres.conf` Datensätze deklarieren. Eine niedrigere Anzahl ist zulässig, bis auf `0`: der `gres.conf` Datensätze beschreiben die Geräte, die auf dem Knoten vorhanden sind, während `Gres` gibt an, wie viele davon der Scheduler zuweisen kann.
- EIN `slurm.conf` `Gres` Ein Eintrag gibt einen GPU-Typ an, in dem kein GPU-Datensatz gespeichert ist `gres.conf` deklariert.

Die letzten beiden Prüfungen hängen davon ab, was Ihre Datensätze deklarieren. AWS PCS kann Geräte nicht zählen, die Slurm beim Booten des Nodes entdeckt, also ein Datensatz, der nur aktiviert `AutoDetect` akzeptiert alle `gpu` zählen Sie und geben Sie das ein `Gres` Eintrag. So bestehen MIG-Profile die Validierung. AWS PCS überspringt auch die Zählprüfung für gemeinsam genutztes GRES (`mps` und `shard`) und überspringt die Typprüfung, wenn ein GPU-Datensatz nein deklariert `Type`.

AWS PCS akzeptiert einen Datensatz, für den es keine Übereinstimmung gibt `Gres` Eintrag. Die Ressource ist auf dem Knoten vorhanden, aber Jobs können sie nicht anfordern. Fügen Sie das hinzu `Gres` Eintrag, um es planbar zu machen.

> **Important**
>
> AWS PCS validiert die Syntax Ihrer Datensätze und deren Konsistenz mit `Gres` Einstellung. Sie werden nicht anhand der Hardware validiert. Ein Rekord, dessen `File`, `Cores`, oder `Links` Das Layout stimmt nicht mit den Geräten auf dem gestarteten Knoten überein, wodurch der Knoten überlastet wird. Das Gleiche gilt für MIG: die Profile, in denen Sie deklarieren `Gres` muss mit den auf dem Knoten konfigurierten Profilen übereinstimmen. Weitere Hinweise zur Diagnose eines überlasteten Knotens finden Sie unter Beheben Sie Fehler bei ungültigen Knotenregistrierungen (INVALID_REG) in AWS STCK.

Slurm-Buchhaltung in AWS 5 STCK

> **Note**
>
> Die Buchhaltung wird für Slurm 24.11 oder höher unterstützt.

Du kannst die Buchhaltung auf deinem neuen aktivieren AWS PCS-Cluster zur Überwachung der Cluster-Nutzung, zur Durchsetzung von Ressourcenbeschränkungen und zur Verwaltung einer detaillierten Zugriffskontrolle für bestimmte Warteschlangen oder Rechenknotengruppen. AWS PCS erstellt und verwaltet die Buchhaltungsdatenbank für Ihren Cluster, sodass Sie keine eigene separate Buchhaltungsdatenbank erstellen und verwalten müssen. AWS PCS verwendet die Buchhaltungsfunktion in Slurm. Weitere Informationen zur Buchhaltungsfunktion in Slurm finden Sie im Slurm-Dokumentation bei SchedMD.

Um Accounting zu verwenden, aktivieren Sie es, wenn Sie einen neuen Cluster erstellen, und legen Sie optional Accounting-Parameter fest. Nachdem Ihr Cluster-Status `Active` und verfügt über Rechenknotengruppen. Sie können sich mit der Linux-Shell eines Anmeldeknotens verbinden, um Abrechnungsfunktionen auszuführen, z. B. das Anzeigen von Jobdaten mit dem Slurm `sacct` Befehl.

AWS PCS console

Auf der Cluster erstellen Seite, unter Scheduler-Einstellungen, aktivieren Buchhaltung.

AWS PCS API

Stellen Sie das zur Verfügung `accounting` Konfiguration in Ihrem Anruf an die `CreateCluster` API-Aktion. In der `accounting` Objekt, stelle das ein `mode` zu `STANDARD`. Weitere Informationen finden Sie unter CreateCluster und Buchhaltung in der AWS PCS-API-Referenz.

Das folgende Beispiel verwendet AWS CLI um den aufzurufen `CreateCluster` API-Aktion. Die Teilzeichenfolge des Parameterwerts `accounting='{mode=STANDARD}'` ermöglicht die Buchhaltung.

```
aws pcs create-cluster --cluster-name cluster-name \
                       --scheduler type=SLURM,version=25.11 \
                       --size SMALL \
                       --networking subnetIds=cluster-subnet-
id,securityGroupIds=cluster-security-group-id \
                       --slurm-configuration
 scaleDownIdleTimeInSeconds=180,accounting='{mode=STANDARD}',slurmCustomSettings='[{paramete
```

> **Important**
>
> Wenn Sie die Buchhaltung aktivieren, fallen zusätzliche Abrechnungsgebühren an. Weitere Informationen finden Sie auf der AWS PCS-Preisseite.

Buchhaltungseinstellungen ändern

Sie können die Kontoführung für vorhandene Cluster aktivieren oder deaktivieren, ohne Ihre Infrastruktur neu aufbauen zu müssen. Weitere Informationen finden Sie unter Aktualisierung eines Clusters in AWS STK..

Wenn Sie das Accounting deaktivieren, wird die Abrechnung für die Accounting-Funktion beendet, sobald der Cluster in den `UPDATING` Bundesstaat. Wenn Sie Accounting aktivieren, beginnt die Abrechnung, wenn der Cluster erfolgreich zum `ACTIVE` Bundesstaat.

Schlüsselkonzepte für die Slurm-Buchhaltung in AWS 5 STCK

Die folgenden Konzepte sind spezifisch für AWS PCS und wie steuern AWS PCS implementiert Slurm Accounting.

Buchhaltungsdatenbank

AWS PCS speichert Ihre Buchhaltungsdaten in einer Datenbank, die in einer erstellt wurde AWS-Konto Das AWS besitzt. Sie haben keinen Zugriff auf `slurmdbd.conf`.

Standardzeit für das Löschen

Das AWS Die PCS-Einstellung gibt die Aufbewahrungsdauer (in Tagen) für alle Arten von Buchhaltungsdatensätzen (Jobs, Ereignisse, Reservierungen, Schritte, Sperrungen, Transaktionen, Nutzungsdaten) an. Wenn der Wert beispielsweise 30 ist, AWS PCS bewahrt Buchhaltungsunterlagen 30 Tage lang auf. Sie geben diesen Wert an, wenn Sie den Cluster erstellen. Wenn Sie keinen Wert angeben, AWS PCS speichert Buchhaltungsunterlagen auf unbestimmte Zeit in der Datenbank.

AWS PCS console

Im Rahmen der Schritte zum Erstellen eines Clusters geben Sie die Standardzeit für das Löschen an. Auf der Cluster erstellen Seite, Buchhaltung aktivieren. Unter Scheduler-Einstellungen, geben Sie einen Ganzzahlwert an für Standardzeit für das Löschen (Tage).

AWS PCS API

Geben Sie die an `defaultPurgeTimeInDays` als Teil der `accounting` Informationen, die Sie in Ihrem Anruf an die `CreateCluster` API-Aktion. Weitere Informationen finden Sie unter CreateCluster und Buchhaltung in der AWS PCS-API-Referenz.

> **Note**
>
> Wenn du den benutzt AWS PCS-API zum Erstellen eines Clusters, der Standardwert für `defaultPurgeTimeInDays` ist `-1` und `0` ist kein gültiger Wert.

Durchsetzung der Rechnungslegungsrichtlinien

Diese Einstellung bestimmt, wie strikt Slurm die Regeln für die Einreichung von Jobs, Ressourcenbeschränkungen und Abrechnungsrichtlinien für Ihren Cluster durchsetzt. Diese Einstellung entspricht der `AccountingStorageEnforce` Parameter in Ihrem Cluster `slurm.conf` datei. Sie können eine beliebige Kombination von Durchsetzungsoptionen auswählen. Wenn Sie keine Optionen auswählen, gelten für Jobs im Cluster keine Buchhaltungsbeschränkungen. AWS PCS unterstützt die folgenden Optionen:

- Verbände — Zuordnung von Job zu Account
- Grenzen — Beschränkungen der Ressourcen
- QoS — Anforderungen an die Servicequalität
- abgesicherter Modus — garantierte Fertigstellung innerhalb bestimmter Grenzen
- keine Schritte — Deaktiviere die Schrittabrechnung
- keine Jobs — Jobabrechnung deaktivieren

Weitere Informationen zu diesen Optionen finden Sie auf der Slurm-Dokumentation bei SchedMD.

AWS PCS console

Sie legen die Optionen im Rahmen der Schritte zum Erstellen eines Clusters fest. Auf der Cluster erstellen Seite, Buchhaltung aktivieren. Wählen Sie die gewünschten Optionen aus dem Durchsetzung der Rechnungslegungsgrundsätze Drop-down-Liste unter Scheduler-Einstellungen.

AWS PCS API

In Slurm werden diese Optionen in einem Cluster festgelegt `slurm.conf` datei. Sie haben keinen direkten Zugriff auf `slurm.conf` für dein AWS PCS-Cluster. Stattdessen stellen Sie zur Verfügung `SlurmCustomSettings` zum `CreateCluster` API-Aktion, wenn Sie einen Cluster erstellen. Weitere Informationen finden Sie unter CreateCluster in der AWS PCS-API-Referenz.

Rufen Sie die Buchhaltungskonfiguration für eine bestehende ab AWS PCS-Cluster

Die Slurm-Buchhaltungskonfiguration ist in der Slurm-Konfiguration für Ihren Cluster enthalten.

AWS PCS console

1. Wählen Sie Cluster aus dem Navigationsbereich.
2. Wählen Sie den Clusternamen aus der Liste aus.
3. Auf der Konfiguration Registerkarte, finden Sie die Buchhaltungskonfiguration unter Slurm-Konfiguration

AWS PCS API

Benutze die `GetCluster` API-Aktion, um die Cluster-Konfiguration abzurufen. Die Accounting-Konfiguration finden Sie in der `slurmConfiguration`. Die Einstellung für `mode` und der Wert von `defaultPurgeTimeInDays` sind unter `accounting`. Die ausgewählten Optionen zur Durchsetzung der Rechnungslegungsgrundsätze finden Sie unter `slurmCustomSettings`. Weitere Informationen finden Sie unter GetCluster in der AWS PCS-API-Referenz.

Setzen Sie die REST-API ein AWS 5 STCK

AWS PCS bietet verwalteten Support für die native REST-API von Slurm über `slurmrestd`, das eine HTTP-Schnittstelle für die programmatische Cluster-Interaktion bereitstellt. Sie können Jobs einreichen, den Clusterstatus überwachen und Ressourcen über Standard-HTTP-Anfragen verwalten, ohne direkten Shell-Zugriff auf Ihren Cluster zu benötigen.

Häufige Anwendungsfälle

Die Slurm REST-API unterstützt verschiedene Integrationsszenarien:

- Integration von Webanwendungen: Erstellen Sie benutzerdefinierte Frontends und Webanwendungen, die Jobs direkt einreichen und verwalten.
- Jupyter Notebook-Integration: Ermöglicht Forschern, Jobs von Notebook-Umgebungen aus einzureichen, ohne ihren Entwicklungsworkflow zu verlassen.
- Integration der Partnerlösung: Verbinden Sie HPC-Tools und Workflow-Manager von Drittanbietern mit Ihrem AWS PCS-Cluster.
- Programmatisches Clustermanagement: Automatisieren Sie Arbeitsabläufe für die Einreichung, Überwachung und Ressourcenverwaltung von Aufträgen.
- Arbeitsabläufe im Bereich Forschung und Datenverarbeitung: Unterstützen Sie akademische und betriebliche Forschungsumgebungen, die Folgendes erfordern API-driven Verwaltung von Aufträgen.

Anforderungen und Einschränkungen

Bevor Sie die Slurm-REST-API verwenden, überprüfen Sie diese Details:

- Ihr Cluster muss Slurm Version 25.05 oder höher verwenden.

- Auf den API-Endpunkt kann nur über eine private IP-Adresse innerhalb der VPC Ihres Clusters zugegriffen werden.
- Ihre Cluster-Sicherheitsgruppe muss HTTP-Verkehr auf Port 6820 zulassen.
- Für die Authentifizierung sind JWT-Token mit bestimmten Ansprüchen an die Benutzeridentität erforderlich.

Zu den aktuellen Einschränkungen gehören:

- Tokens generiert von `scontrol token` werden nicht unterstützt.
- `X-SLURM-USER-NAME` Header-Imitation ist nicht verfügbar.
- Für einige Funktionen muss das Slurm-Konto aktiviert sein.
- Nicht kompatibel mit dem Slurm-CLI-Filter-Plugin-Mechanismus.
- Verbindungen zum REST-API-Endpunkt sind nicht mit TLS verschlüsselt.

Themen

- Aktivierung der Slurm-REST-API in AWS 5 STCK
- Authentifizierung mit der Slurm-REST-API in AWS 5 STCK
- Verwendung der Slurm-REST-API für die Auftragsverwaltung in AWS 5 STCK
- Häufig gestellte Fragen zur Slurm-REST-API in AWS STCK

Aktivierung der Slurm-REST-API in AWS 5 STCK

Aktivieren Sie die Slurm-REST-API, um auf die HTTP-Schnittstelle Ihres Clusters für die programmatische Auftragsverwaltung und -überwachung zuzugreifen. Sie können diese Funktion bei der Cluster-Erstellung aktivieren oder einen vorhandenen Cluster aktualisieren, der die Anforderungen erfüllt.

Voraussetzungen

Bevor Sie die Slurm-REST-API aktivieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Cluster-Version: Slurm-Version 25.05 oder höher.
- Sicherheitsgruppe: Regeln, die HTTP-Verkehr auf Port 6820 aus den von Ihnen gewünschten Quellen zulassen.

Verfahren

Um die Slurm REST-API auf einem neuen Cluster zu aktivieren

AWS-Managementkonsole

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Wählen Sie Cluster erstellen.
3. Unter Einzelheiten zum Cluster, wählen Sie Slurm Version 25.05 oder höher.
4. Konfigurieren Sie die anderen Cluster-Einstellungen nach Bedarf.
5. In der Scheduler-Konfiguration Abschnitt, Satz API TESTEN zu Aktiviert.
6. Konfigurieren Sie Ihre Cluster-Sicherheitsgruppe so, dass HTTP-Verkehr auf Port 6820 von Ihren gewünschten Quellen zugelassen wird.
7. Schließen Sie den Cluster-Erstellungsprozess ab.

AWS CLI

1. Fügen Sie bei der Erstellung Ihres Clusters eine Slurm-REST-Konfiguration hinzu.

```
aws pcs create-cluster --region region \
    --cluster-name my-cluster \
    --scheduler type=SLURM, version=25.05 \
    --size SMALL \
    --networking subnetIds=subnet-ExampleId1,securityGroupIds=sg-ExampleId1 \
    --slurm-configuration slurmRest='{mode=STANDARD}'
```

2. Konfigurieren Sie Ihre Cluster-Sicherheitsgruppe so, dass HTTP-Verkehr auf Port 6820 von Ihren gewünschten Quellen zugelassen wird.

Um die Slurm-REST-API auf einem vorhandenen Cluster zu aktivieren

AWS-Managementkonsole

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Wählen Sie Ihren Cluster aus der Liste aus.
3. Vergewissern Sie sich, dass Ihr Cluster in den Cluster-Details Slurm Version 25.05 oder höher verwendet.

4. Wählen Sie Cluster bearbeiten.
5. In der Scheduler-Konfiguration Abschnitt, Satz API TESTEN zu Aktiviert.
6. Wählen Cluster aktualisieren um die Änderungen anzuwenden.
7. Konfigurieren Sie Ihre Cluster-Sicherheitsgruppe so, dass HTTP-Verkehr auf Port 6820 von Ihren gewünschten Quellen zugelassen wird.

AWS CLI

1. Aktualisieren Sie Ihren Cluster mit einer Slurm-REST-Konfiguration, wie in diesem Beispiel.

```
aws pcs update-cluster --cluster-identifier my-cluster \
    --slurm-configuration 'slurmRest={mode=STANDARD}'
```

2. Konfigurieren Sie Ihre Cluster-Sicherheitsgruppe so, dass HTTP-Verkehr auf Port 6820 von Ihren gewünschten Quellen zugelassen wird.

Was passiert nach der Aktivierung

Wenn Sie die REST-API aktivieren, AWS PCS automatisch:

- Generiert einen JWT-Signaturschlüssel und speichert ihn in AWS Manager für Geheimnisse.
- Macht den API-Endpunkt verfügbar unter `http://<clusterPrivateIpAddress>:6820` innerhalb Ihrer VPC.
- Aktualisiert Ihre Cluster-Konfiguration, um die Details des REST-API-Endpunkts anzuzeigen.

Sie können sich jetzt authentifizieren und die REST-API für die Auftragsverwaltung und Clustervorgänge verwenden.

Authentifizierung mit der Slurm-REST-API in AWS 5 STCK

Die Slurm-REST-API in AWS PCS verwendet die JSON Web Token (JWT) -Authentifizierung, um den sicheren Zugriff auf Ihre Cluster-Ressourcen zu gewährleisten. AWS PCS bietet einen verwalteten Signaturschlüssel, der in gespeichert ist AWS Secrets Manager, mit dem Sie JWT-Token generieren, die bestimmte Ansprüche an die Benutzeridentität enthalten.

Voraussetzungen

Bevor Sie sich mit der Slurm-REST-API authentifizieren, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Cluster-Konfiguration: AWS PCS-Cluster mit Slurm 25.05+ und aktivierter REST-API.
- AWS-Berechtigungen: Zugriff auf AWS Secrets Manager für den JWT-Signaturschlüssel.
- Informationen zum Benutzer: Benutzername, POSIX-Benutzer-ID und eine oder mehrere POSIX-Gruppen-IDs für Ihr Clusterkonto.
- Netzwerkzugriff: Konnektivität innerhalb der VPC Ihres Clusters mit Sicherheitsgruppe, die Port 6820 zulässt.

Verfahren

Um die Slurm-REST-API-Endpunktadresse abzurufen

AWS-Managementkonsole

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Wählen Sie Ihren Cluster aus der Liste aus.
3. Suchen Sie in den Cluster-Konfigurationsdetails nach dem Endpunkte Abschnitt.
4. Notieren Sie sich die private IP-Adresse und den Port für Slurm REST-API (slurmrestd).
5. Sie können API-Aufrufe tätigen, indem Sie ordnungsgemäß formatierte HTTP-Anfragen an diese Adresse senden.

AWS CLI

1. Fragen Sie Ihren Cluster-Status mit ab `aws pcs get-cluster`. Suchen Sie nach dem `SLURMRESTD` Endpunkt im `endpoints` Feld in der Antwort. Ein Beispiel:

```
"endpoints": [
      {
          "type": "SLURMCTLD",
          "privateIpAddress": "192.0.2.1",
          "port": "6817"
      },
      {
          "type": "SLURMRESTD",
```

```
        "privateIpAddress": "192.0.2.1",
        "port": "6820"
    }
]
```

2. Sie können API-Aufrufe tätigen, indem Sie korrekt formatierte HTTP-Anfragen an senden `http://<privateIpAddress>:<port>/`

Um den JWT-Signaturschlüssel abzurufen

1. Öffnen Sie den AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Wählen Sie Ihren Cluster aus der Liste aus.
3. Suchen Sie in den Cluster-Konfigurationsdetails nach dem Scheduler-Authentifizierung Abschnitt.
4. Beachten Sie die JSON Web Token (JWT) -Schlüssel ARN und Version.
5. Benutze den AWS CLI um den Signaturschlüssel von Secrets Manager abzurufen:

```
aws secretsmanager get-secret-value --secret-
id arn:aws:secretsmanager:region:account:secret:name --version-id version
```

Um ein JWT-Token zu generieren

1. Erstellen Sie ein JWT mit den folgenden erforderlichen Ansprüchen:
   - `exp` — Ablaufzeit in Sekunden seit 1970 für das JWT
   - `iat` — Aktuelle Zeit in Sekunden seit 1970
   - `uid` — Die POSIX-Benutzer-ID
   - `gid` — Die POSIX-Gruppen-ID
   - `id` — POSIX-Identitätseigenschaften
     - `name` — Der Benutzername für die Authentifizierung
     - `gecos` — Benutzerkommentarfeld, das häufig zum Speichern eines menschenlesbaren Namens verwendet wird
     - `dir` — Home-Verzeichnis des Benutzers
     - `shell` — Standard-Shell des Benutzers
     - `gids` — Liste zusätzlicher POSIX-Gruppen-IDs, in denen sich der Benutzer befindet

2. Signieren Sie das JWT mit dem von Secrets Manager abgerufenen Signaturschlüssel.
3. Stellen Sie eine angemessene Ablaufzeit für das Token ein.

> **Note**
>
> Geben Sie den Benutzernamen in der `name` Feld innerhalb von `id` Anspruch. AWS PCS akzeptiert diesen Anspruch für jede unterstützte Slurm-Version.
> Sie können den Benutzernamen auch in einem der folgenden Claims auf oberster Ebene festlegen:
>
> - `sun`
> - `username`
> - Ein benutzerdefinierter Forderungsname, den Sie mit definieren `userclaimfield` in der `AuthAltParameters` Benutzerdefinierte Slurm-Einstellung
>
> Slurm 26.05 liest den Benutzernamen aus dem `name` Feld innerhalb des `id` Anspruch.
> Die Ansprüche der obersten Ebene sind ab Slurm 26.05 veraltet, und AWS PCS könnte sie in einer zukünftigen Slurm-Version nicht mehr akzeptieren. Bestehende Token, die einen Anspruch auf oberster Ebene verwenden, authentifizieren sich weiterhin auf allen unterstützten Versionen. Wenn Ihre Token einen verwenden, aktualisieren Sie den Code, der sie generiert, um den Benutzernamen in der `name` Feld innerhalb der `id` Anspruch.
> Geben Sie den Benutzernamen nur an einer Stelle ein. Wenn ein Token beide festlegt `name` Feld innerhalb der `id` Claim und ein Claim auf oberster Ebene, der Anspruch, der Vorrang hat, hängt von der Slurm-Version Ihres Clusters ab: auf Slurm 26.05 und höher ist der `name` Feld innerhalb von `id` Anspruch hat Vorrang. In früheren Versionen hat der Anspruch der obersten Ebene Vorrang.

Um API-Anfragen zu authentifizieren

1. Fügen Sie das JWT-Token mit einer der folgenden Methoden in Ihre HTTP-Anfragen ein:
   - Inhaber-Token — Hinzufügen `Authorization: Bearer <jwt>` Kopfzeile
   - Slurm-Header — Hinzufügen `X-SLURM-USER-TOKEN: <jwt>` Kopfzeile
2. Stellen Sie HTTP-Anfragen an den REST-API-Endpunkt:

Hier ist ein Beispiel für den Zugriff auf `/ping` API mit Curl und `Authorized: Bearer` Kopfzeile.

```
curl -X GET -H "Authorization: Bearer <jwt>" \
      http://<privateIpAddress>:6820/slurm/v0.0.43/ping
```

Beispiel für eine JWT-Generierung

Holen Sie sich das AWS PCS-Cluster-JWT-Signaturschlüssel und speichern Sie ihn als lokale Datei. Ersetzen Sie die Werte für aws-region, geheime Arne, und geheime Version mit Werten, die für Ihren Cluster geeignet sind.

```
#!/bin/bash
SECRET_KEY=$(aws secretsmanager get-secret-value \
  --region aws-region \
  --secret-id secret-arn \
  --version-id secret-version \
  --query 'SecretString' \
  --output text)
echo "$SECRET_KEY" | base64 --decode > jwt.key
```

Dieses Python-Beispiel veranschaulicht, wie der Signaturschlüssel verwendet wird, um ein JWT-Token zu generieren:

```
#!/usr/bin/env python3

import sys
import os
import pprint
import json
import time
from datetime import datetime, timedelta, timezone
from jwt import JWT
from jwt.jwa import HS256
from jwt.jwk import jwk_from_dict
from jwt.utils import b64decode,b64encode
if len(sys.argv) != 3:
    sys.exit("Usage: gen_jwt.py [jwt_key_file] [expiration_time_seconds]")
SIGNING_KEY = sys.argv[1]
EXPIRATION_TIME = int(sys.argv[2])
```

```
with open(SIGNING_KEY, "rb") as f:
    priv_key = f.read()
signing_key = jwk_from_dict({
    'kty': 'oct',
    'k': b64encode(priv_key)
})
message = {
    "exp": int(time.time() + EXPIRATION_TIME),
    "iat": int(time.time()),
    "uid": 1000,
    "gid": 1000,
    "id": {
        "name": "ec2-user",
        "gecos": "EC2 User",
        "dir": "/home/ec2-user",
        "gids": [1000],
        "shell": "/bin/bash"
    }
}
a = JWT()
compact_jws = a.encode(message, signing_key, alg='HS256')
print(compact_jws)
```

Das Skript druckt ein JWT auf den Bildschirm.

```
abcdefgtjwttoken...
```

Verwendung der Slurm-REST-API für die Auftragsverwaltung in AWS 5 STCK

Überblick über die Slurm REST-API

Die Slurm REST-API bietet programmatischen Zugriff auf Cluster-Management-Funktionen über HTTP-Anfragen. Wenn Sie diese Hauptmerkmale verstehen, können Sie die API effektiv nutzen mit AWS STCK.:

- Zugriffs-Protokoll: Die API verwendet HTTP (nicht HTTPS) für die Kommunikation innerhalb des privaten Netzwerks Ihres Clusters.
- Angaben zur Verbindung: Greifen Sie mit der privaten IP-Adresse Ihres Clusters auf die API zu und `slurmrestd` Port (normalerweise 6820). Das vollständige Basis-URL-Format lautet `http://<privateIpAddress>:6820`.

- API-Versionierung: Die API-Version entspricht Ihrer Slurm-Installation. Verwenden Sie für Slurm 25.05 die Version v0.0.43. Die Versionsnummer ändert sich mit jeder Slurm-Version. Die aktuell unterstützten API-Versionen finden Sie in der Versionshinweise zu Slurm.
- URL-Struktur: Die URL-Struktur für die Slurm-REST-API lautet `http://<privateIpAddress>:<port>/<api-version>/<endpoint>`. Detaillierte Nutzungsinformationen für REST-API-Endpunkte finden Sie im Slurm-Dokumentation.

Spezifische Informationen zur Arbeit mit der Slurm-REST-API finden Sie im REST-Client Slurm-Dokumentation.

Voraussetzungen

Bevor Sie die Slurm-REST-API verwenden, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Cluster-Konfiguration: AWS PCS-Cluster mit Slurm 25.05+ und aktivierter REST-API.
- Authentifizierung: Gültiges JWT-Token mit korrekten Ansprüchen an die Benutzeridentität.
- Netzwerkzugriff: Konnektivität innerhalb der VPC Ihres Clusters mit einer Sicherheitsgruppe, die Port 6820 zulässt.

Verfahren

Um einen Job mithilfe der REST-API einzureichen

1. Erstellen Sie eine Anfrage zur Einreichung eines Jobs mit den erforderlichen Parametern:

```
{
  "job": {
    "name": "my-job",
    "partition": "compute",
    "nodes": 1,
    "tasks": 1,
    "script": "#!/bin/bash\necho 'Hello from Slurm REST API'",
    "environment": ["PATH=/usr/local/bin:/usr/bin:/bin"]
  }
}
```

2. Senden Sie den Job mit einer HTTP-POST-Anfrage:

```
curl -X POST \
```

```
  -H "Authorization: Bearer <jwt>" \
  -H "Content-Type: application/json" \
  -d '<job-json>' \
  http://<privateIpAddress>:6820/slurm/v0.0.43/job/submit
```

3. Notieren Sie sich die Job-ID, die in der Antwort zu Überwachungszwecken zurückgegeben wurde.

Um den Auftragsstatus zu überwachen

1. Informieren Sie sich über einen bestimmten Job:

```
curl -X GET -H "Authorization: Bearer <jwt>" \
    http://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

2. Listet alle Jobs für den authentifizierten Benutzer auf:

```
curl -X GET -H "Authorization: Bearer <jwt>" \
    http://<privateIpAddress>:6820/slurm/v0.0.43/jobs
```

So brechen Sie einen Auftrag ab

- Senden Sie eine DELETE-Anfrage, um einen bestimmten Job abzubrechen:

```
curl -X DELETE -H "Authorization: Bearer <jwt>" \
    http://<privateIpAddress>:6820/slurm/v0.0.43/job/<job-id>
```

Häufig gestellte Fragen zur Slurm-REST-API in AWS STCK

Dieser Abschnitt beantwortet häufig gestellte Fragen zur Slurm-REST-API in AWS STCK.

Was ist die Slurm-REST-API?

Die Slurm REST-API ist eine HTTP-Schnittstelle, die es Ihnen ermöglicht, programmgesteuert mit dem Slurm Workload Manager zu interagieren. Sie können Standard-HTTP-Methoden wie GET, POST und DELETE verwenden, um Jobs einzureichen, den Cluster-Status zu überwachen und Ressourcen zu verwalten, ohne dass ein Befehlszeilenzugriff auf den Cluster erforderlich ist.

Kann ich Tokens verwenden, die von generiert wurden **`scontrol token`**?

Nein, Standard `scontrol token` Die Ausgabe ist nicht kompatibel mit AWS PCS auf jeder unterstützten Slurm-Version, einschließlich 26.05. Die PCS Slurm REST-API erfordert erweiterte JWT-Token, die POSIX-Identitätsansprüche enthalten, und `scontrol token` generiert ein Token ohne sie. Tokens, denen die erforderlichen Ansprüche fehlen, werden von der API abgelehnt. Informationen zu den Ansprüchen, die ein Token enthalten muss, finden Sie unter Authentifizierung mit der Slurm-REST-API in AWS 5 STCK.

Kann ich von außerhalb meiner VPC auf die API zugreifen?

Nein, der REST-API-Endpunkt ist nur von Ihrer VPC aus über die private IP-Adresse des Slurm-Controllers zugänglich. Um den externen Zugriff zu ermöglichen, implementieren Sie AWS Dienste wie Application Load Balancer mit VPC Link, API Gateway oder richten Sie VPC-Peering- oder VPN-Verbindungen für sichere Konnektivität ein.

Warum verwendet die API HTTP statt HTTPS?

Die Slurm-REST-API soll ein interner Endpunkt im privaten Netzwerk Ihres Clusters sein. Für Produktionsbereitstellungen, die eine Verschlüsselung erfordern, können Sie Folgendes implementieren SSL/TLS Terminierung auf einer höheren Ebene in Ihrer Architektur, z. B. über ein API-Gateway, einen Load Balancer oder einen Reverse-Proxy.

Wie kontrolliere ich den Zugriff auf die REST-API?

Konfigurieren Sie die Sicherheitsgruppenregeln Ihres Clusters, um den Zugriff auf Port 6820 auf dem Slurm-Controller einzuschränken. Legen Sie Regeln für eingehenden Verkehr fest, um nur Verbindungen von vertrauenswürdigen IP-Bereichen oder bestimmten Quellen innerhalb Ihrer VPC zuzulassen und so den unbefugten Zugriff auf den API-Endpunkt zu blockieren.

Wie rotiere ich den JWT-Signaturschlüssel?

Versetzen Sie Ihren Cluster in den Wartungsmodus ohne aktive Instanzen und leiten Sie dann die Schlüsselrotation wie folgt ein AWS Verwaltung von Geheimnissen. Aktivieren Sie die Warteschlangen nach Abschluss der Rotation erneut. Alle vorhandenen JWT-Token werden ungültig und müssen mit dem neuen Signaturschlüssel von Secrets Manager neu generiert werden.

Muss das Slurm-Konto aktiviert sein, um die REST-API verwenden zu können?

Nein, die Slurm-Buchhaltung ist für grundlegende REST-API-Operationen wie die Einreichung und Überwachung von Aufträgen nicht erforderlich. Allerdings das gesamte `/slurmdb` Für den Endpunkt muss die Buchhaltung aktiv sein.

Welche Tools von Drittanbietern funktionieren mit AWS PCS REST-API?

Mit vielen bestehenden Slurm-REST-API-Clients sollten Sie arbeiten AWS PCS, einschließlich Slurm Exporter für Prometheus, SlurmWebund benutzerdefinierte Anwendungen, die dem Standardformat der Slurm-REST-API folgen. Tools, die sich jedoch auf Folgendes verlassen `scontrol token` für die Authentifizierung müssen Änderungen vorgenommen werden, damit sie verwendet werden können AWS PCS JWT-Anforderungen.

Gibt es zusätzliche Kosten für die Nutzung der REST-API?

Nein, es fallen keine zusätzlichen Gebühren für die Aktivierung oder Nutzung der Slurm REST-API-Funktion an. Sie zahlen wie gewohnt nur für die zugrunde liegenden Cluster-Ressourcen.

Wie kann ich Probleme mit der REST-API beheben?

- Probleme mit der Netzwerkkonnektivität

  Wenn Sie den API-Endpunkt nicht erreichen können, treten bei HTTP-Anfragen an den Cluster-Controller Zeitüberschreitungen oder Fehler „Verbindung verweigert“ auf.

  Was ist zu tun: Stellen Sie sicher, dass sich Ihr Client in derselben VPC befindet oder über ein korrektes Netzwerkrouting verfügt, und stellen Sie sicher, dass Ihre Sicherheitsgruppe HTTP-Verkehr auf Port 6820 von Ihrer Quell-IP oder Ihrem Subnetz zulässt.

- Probleme mit der Slurm-REST-Authentifizierung

  Wenn Ihr JWT-Token ungültig, abgelaufen oder falsch signiert ist, geben API-Anfragen im Fehlerfeld der Antwort „Protokollauthentifizierungsfehler“ zurück.

  Beispiel für eine Fehlermeldung:

```
{
"errors": [
    {
    "description": "Batch job submission failed",
    "error_number": 1007,
    "error": "Protocol authentication error",
    "source": "slurm_submit_batch_job()"
    }
  ]
}
```

Was ist zu tun: Stellen Sie sicher, dass Ihr JWT-Token richtig formatiert, nicht abgelaufen und mit dem richtigen Schlüssel von Secrets Manager signiert ist. Stellen Sie sicher, dass das Token ordnungsgemäß formatiert ist und die erforderlichen Ansprüche enthält und dass Sie das richtige Header-Format für die Authentifizierung verwenden.

- Der Auftrag kann nach dem Absenden nicht ausgeführt werden

  Wenn Ihr JWT-Token gültig ist, aber eine falsche interne Struktur oder einen falschen Inhalt enthält, wurde für Jobs möglicherweise ein pausiertes (PD) Bundesstaat mit Ursachencode `JobAdminHead`. Benutzen `scontrol show job <job-id>` um den Job zu inspizieren — du wirst sehen `JobState=PENDING, Reason=JobHeldAdmin`, und `SystemComment=slurm_cred_create failure, holding job`.

  Was ist zu tun: Die Hauptursache können falsche Werte in JWT sein. Stellen Sie sicher, dass das Token ordnungsgemäß strukturiert ist und die erforderlichen Ansprüche gemäß der PCS-Dokumentation enthält.

- Probleme mit den Zugriffsrechten für das Arbeitsverzeichnis

  Wenn die in Ihrem JWT angegebene Benutzeridentität keine Schreibberechtigungen für das Arbeitsverzeichnis des Jobs hat, schlägt der Job mit Berechtigungsfehlern fehl, ähnlich wie bei `sbatch --chdir` mit einem Verzeichnis, auf das nicht zugegriffen werden kann.

  Was ist zu tun: Stellen Sie sicher, dass der in Ihrem JWT-Token angegebene Benutzer über die entsprechenden Berechtigungen für das Arbeitsverzeichnis des Jobs verfügt.

- Haben Sie immer noch Probleme?

  1. Überprüfen Sie die von SchedMD Dokumentation auf der REST-API-Spezifikation.
  2. Weitere Informationen zu Fehlern finden Sie in den Slurm-Controller-Protokollen (siehe Scheduler loggt sich in AWS PCS ein für weitere Details).

Rechenknoten mit Slurm neu starten AWS 5 STÜCK

AWS PCS unterstützt Slurms native Version `scontrol reboot` Befehl. Verwenden Sie diesen Befehl, um Rechenknoten neu zu starten, ohne die EC2-Instanz zu ersetzen. Andere Neustartmethoden (Amazon EC2-Konsole, AWS CLI, automatische Patches oder Systemwartung) verursachen AWS PCS, um die EC2-Instance als fehlerhaft einzustufen und zu ersetzen.

Vorteile des Slurm-Neustarts

Der Slurm-Neustart bietet mehrere Vorteile für die Cluster-Wartung:

- Kapazität beibehalten — Vermeiden Sie den Verlust von EC2-Instances mit eingeschränkter Kapazität an andere Kunden.
- Reduzieren Sie die Kosten — Vermeiden Sie unnötige Zyklen für den Austausch von Instanzen und die fortlaufende Abrechnung inaktiver Knoten.
- Schnellere Wiederherstellung — Keine Verzögerungen bei der Bereitstellung im Vergleich zum Austausch einer Instanz.
- Betriebliche Flexibilität — Beseitigen Sie Speicherlecks, entfernen Sie temporäre Dateien und stellen Sie Knoten aus einem heruntergekommenen Zustand wieder her.

Wann sollte der Slurm-Neustart verwendet werden

Verwenden Sie den Slurm-Neustart für gängige betriebliche Wartungsszenarien:

- Problembehebung — Beheben Sie Leistungsprobleme oder nicht reagierende Prozesse, insbesondere bei GPU-Knoten.
- Bereinigung der Ressourcen — Beseitigt Speicherlecks, temporäre Dateien `/tmp`oder festgefahrene Prozesse, die sich auf die Arbeitsleistung auswirken.
- Wiederherstellung — Stellen Sie Knoten wieder her, wenn sie nicht mehr funktionieren oder heruntergefahren sind, bevor ein vollständiger Austausch der Knoten erforderlich ist.

Einschränkungen

- Nur Slurm Admin-Benutzer (Root-Benutzer) können Reboot-Befehle ausführen.
- Die Unterstützung für Neustarts ist beschränkt auf `scontrol reboot` nur.
- RebootProgram Konfiguration wird nicht unterstützt.
- Keine Konsolenschnittstelle — nur Befehlszeile.

Themen

- Starten Sie einen Rechenknoten mit Slurm in neu AWS 5 STCK
- Stornieren Sie einen ausstehenden Neustart in AWS STCK

- Häufig gestellte Fragen zum Slurm-Neustart in AWS STCK
- Behebung von Problemen beim Slurm-Neustart in AWS STCK

Starten Sie einen Rechenknoten mit Slurm in neu AWS 5 STCK

Verwenden Sie den nativen Reboot-Befehl von Slurm, um Leistungsprobleme zu lösen, Ressourcenprobleme zu beheben oder sich aus einem heruntergekommenen Zustand ohne Verlust der EC2-Instance-Kapazität zu erholen.

Voraussetzungen

- Slurm Admin-Rechte (Root-Benutzerzugriff)
- Zugriff auf einen Login-Knoten im AWS PCS-Cluster

Verfahren

1. Stellen Sie über die EC2-Konsole eine Verbindung zu einem Anmeldeknoten her.
   a. Wählen Sie in der EC2-Konsole Instances aus.
   b. Wählen Sie Ihre Anmeldeknoteninstanz aus.
   c. Wählen Sie Connect aus.
2. Identifizieren Sie den Namen des Ziel-Rechenknotens mit `sinfo` oder `scontrol show node`.

```
sinfo
# or
scontrol show node
```

3. Führen Sie den Reboot-Befehl mit einer der folgenden Optionen aus:

> **Warning**
>
> Nicht verwenden `nextstate=DOWN` mit dem `scontrol reboot` Befehl. Dieser Parameter markiert den Knoten als fehlerhaft und löst die Ersetzung der Instanz aus.

   - Grundlegender Neustart (wartet, bis der Knoten inaktiv wird):

```
scontrol reboot nodename
```

- Sofortiger Neustart (entleert den Knoten und startet neu, wenn die Jobs abgeschlossen sind):

```
scontrol reboot ASAP nodename
```

- Aus gutem Grund neu starten:

```
scontrol reboot ASAP reason="troubleshooting" nodename
```

- Neustart mit Resume-Status:

```
scontrol reboot ASAP nextstate=RESUME nodename
```

4. Überwachen Sie den Fortschritt des Neustarts mit `scontrol show node`.

```
scontrol show node nodename
```

5. Stellen Sie sicher, dass der Knoten nach Abschluss des Neustarts wieder betriebsbereit ist.

Stornieren Sie einen ausstehenden Neustart in AWS STCK

Brechen Sie einen ausstehenden Neustart ab, um unnötige Ausfallzeiten zu vermeiden, wenn das Problem behoben wurde oder der Neustart nicht mehr erforderlich ist.

Voraussetzungen

- Slurm Administratorrechte
- Der Node muss einen ausstehenden Neustart haben (mit dem Status „Neustart ausgeführt“)
- Zugriff auf den Anmeldeknoten für die Befehlsausführung

Verfahren

1. Stellen Sie eine Verbindung zum Anmeldeknoten her.
2. Stellen Sie sicher, dass ein Neustart des Knotens aussteht, indem Sie `scontrol show node`.

```
scontrol show node nodename
```

Suchen Sie im Knotenstatus nach „Neustart ausgegeben“.

3. Führen Sie den Befehl Abbrechen aus.

```
scontrol cancel_reboot nodename
```

4. Stellen Sie sicher, dass der Neustart abgebrochen wurde und der Knotenstatus wieder normal ist.

```
scontrol show node nodename
```

Häufig gestellte Fragen zum Slurm-Neustart in AWS STCK

Antworten auf häufig gestellte Fragen zur Verwendung des Slurm-Neustarts finden Sie in AWS 5 STCK.

Was ist die Unterstützung für Slurm-Neustarts?

Unterstützung für den nativen Slurm `scontrol reboot` Befehl. Verwenden Sie diesen Befehl, um Rechenknoten ohne automatischen Instance-Austausch neu zu starten, wodurch die EC2-Instance-Kapazität erhalten bleibt und die Betriebskosten gesenkt werden.

Wer kann Slurm-Reboot-Befehle verwenden?

Nur Slurm Admin-Benutzer (Root-Benutzer) können Reboot-Befehle ausführen. Normale Benutzer versuchen es zu benutzen `scontrol reboot` wird von Slurm die Fehlermeldung „Erlaubnis verweigert“ erhalten, ohne dass dies Auswirkungen auf den Knoten hat.

Was passiert mit laufenden Jobs während eines Neustarts?

Standardmäßig werden Jobs normal abgeschlossen, bevor ein Neustart erfolgt. Bei der Option ASAP wird der Knoten entleert, um neue Jobs zu verhindern. Der Neustart erfolgt nach Abschluss der aktuellen Jobs. Jobs können storniert oder für sofortige Neustarts in die Warteschlange gestellt werden.

Wie unterscheidet sich das vom Neustart der EC2-Konsole?

Bei einem Slurm-Neustart bleibt die EC2-Instance erhalten und ein Austausch wird vermieden. Bei einem Neustart der EC2-Konsole ersetzt PCS die Instance aufgrund fehlgeschlagener Integritätsprüfungen während des Neustarts.

Kann ich benutzerdefinierte Neustartskripte konfigurieren?

Nein, RebootProgram Die Konfiguration wird in der ersten Version nicht unterstützt. Die Funktion verwendet das standardmäßige Slurm-Neustartverhalten ohne Unterstützung für benutzerdefinierte Skripte.

Wie lange dauert ein Slurm-Neustart?

Die Neustartzeit variiert je nach Instance-Typ, Startprozessen des Kunden, AMI-Konfiguration und ob die Jobs zuerst abgeschlossen werden müssen. Der Prozess umfasst das Warten auf den Abschluss der Jobs, einen physischen Neustart, Zustandsprüfungen und die Registrierung des Slurmd-Daemons.

Kann ich eine Historie der Neustarts sehen?

Neustart-Ereignisse werden in Slurm-Logs (slurmctld und slurmd) aufgezeichnet, die wie folgt überwacht werden können CloudWatch. Das Feld „Grund“ im Knotenstatus zeigt den Grund für den Neustart während des Vorgangs an.

Was passiert, wenn ein Knoten beim Neustart hängen bleibt?

Wenn ein Knoten den Neustartvorgang innerhalb von ResumeTimeout, er wird als DOWN markiert. Überprüfe CloudWatch Logs auf Fehler, verifizieren Sie die Netzwerkkonnektivität und untersuchen Sie die Slurmd-Logs. Kontakt AWS Support, falls die Probleme weiterhin bestehen.

Kann ich mehrere Knoten gleichzeitig neu starten?

Ja, Sie können im Reboot-Befehl mehrere Knoten angeben:

```
scontrol reboot ASAP node1,node2,node3
```

Wie kann ich einen Knoten neu starten, ohne auf den Abschluss der Jobs zu warten?

Bei Problemen wie problematischen Knoten, die Jobs mit mehreren Knoten beeinträchtigen, erhebliche Leistungseinbußen oder instabiles GPU-Verhalten auftreten, haben Sie zwei Möglichkeiten, um Knoten sofort neu zu starten:

- Abbrechen und neu starten — Stornieren Sie zunächst die betroffenen Jobs mit `scancel <job_id>`, dann initiieren Sie einen sofortigen Neustart mit `scontrol reboot ASAP <nodename>`. Laufende Jobs werden beendet und müssen nach der Wiederherstellung des Knotens erneut eingereicht werden.
- Drain und Requeue (weniger wirkungsvoll) — Initiieren Sie zunächst einen Drain und starten Sie den Computer neu mit `scontrol reboot ASAP <nodename>`, und stellen Sie dann die

betroffenen Jobs erneut in die Warteschlange mit `scontrol requeue <job_id>`. Dadurch werden Jobs wieder in den Status „Ausstehend“ versetzt, anstatt sie zu stornieren.

Was passiert, wenn ich nextState=down angebe?

Wenn du spezifizierst `nextstate=DOWN`, wird der Knoten nach dem Neustart als fehlerhaft markiert und löst den Austausch der Instanz aus. Um einen Instanzwechsel zu vermeiden, geben Sie nicht nextstate an oder verwenden Sie `nextstate=RESUME`.

Weitere Ressourcen

- Grundlegende Verfahren zum Neustarten finden Sie unter Starten Sie einen Rechenknoten mit Slurm in neu AWS 5 STCK.
- Informationen zur Behebung von Problemen beim Neustart finden Sie unter Behebung von Problemen beim Slurm-Neustart in AWS STCK.
- Die Dokumentation zum Slurm-Neustart finden Sie unter Slurm-Control-Dokumentation.

Behebung von Problemen beim Slurm-Neustart in AWS STCK

Wenn Probleme beim Neustarten des Knotens auftreten, überprüfen Sie zunächst den Knotenstatus mit `scontrol show node` *`nodename`*. Dann untersuche CloudWatch Protokolle sowohl für Slurm (slurmctld und slurmd) als auch für Systemprotokolle, um potenzielle Fehler zu identifizieren.

Überprüfen Sie zur grundlegenden Problembehandlung die Netzwerkkonnektivität, überprüfen Sie die Sicherheitsgruppeneinstellungen und stellen Sie sicher, dass alle erforderlichen Dienste nach dem Neustart ausgeführt werden. Wenn die Probleme nach den grundlegenden Schritten zur Fehlerbehebung weiterhin bestehen, wenden Sie sich an AWS Unterstützung. Wenn Sie sich an den Support wenden, geben Sie relevante Protokollauszüge, Informationen zum Knotenstatus und einen Zeitplan für den Neustartversuch an, um den Problemlösungsprozess zu beschleunigen.

Weitere Ressourcen

- Zur Überwachung AWS PCS-Instanzen verwenden CloudWatch, siehe Überwachung AWS PCS-Instanzen, die Amazon verwenden CloudWatch.
- Allgemeine Hinweise zur Problembehandlung finden Sie unter Behebung von Problemen in AWS Dienst für parallele Datenverarbeitung.
- Die Slurm-Dokumentation finden Sie unter Leitfaden zur Slurm-Fehlerbehebung.

Konfiguration der Hardwaretopologie in AWS STCK

Slurm trifft bessere Platzierungsentscheidungen, wenn es das interne Hardware-Layout eines Rechenknotens kennt: wie sich Kerne zu Sockets gruppieren und Non-Uniform Speicherzugriffsdomänen (NUMA), wie L3-Caches den Kernen zugeordnet werden und an welche Kerne jede GPU angeschlossen ist. In diesem Thema wird beschrieben, wie Slurm dieses Layout modelliert und wie AWS PCS konfiguriert es auf jeder unterstützten Slurm-Version.

CPU-Topologie

Die Kerne eines Rechenknotens sind nicht austauschbar. Sie sind in physische Pakete (Sockets) und NUMA-Domänen gruppiert, und jede Gruppe teilt sich Ressourcen wie Speichercontroller und L3-Cache. Zwei Aufgaben in derselben Gruppe kommunizieren schneller und teilen sich den Cache. Gruppenübergreifende Aufgaben konkurrieren weniger um die gemeinsam genutzten Ressourcen.

Slurm beschreibt diese Gruppierung pro Knoten mit `Sockets`, `CoresPerSocket`, und `ThreadsPerCore` Einstellungen in `slurm.conf`. Was Slurm behandelt als Steckdose ist mit dem pro Knoten konfigurierbar Parameters Einstellung:

- `numa_node_as_socket` — jede NUMA-Domain ist ein Socket.
- `l3cache_as_socket` — jede L3-Cache-Domain ist ein Socket. Dies kann auf bestimmten Prozessoren nützlich sein, bei denen die Socket-Ebene zu grob ist und der L3-Cache möglicherweise eine bessere Aufgabenverteilung bietet.

Wenn die CPU-Topologie konfiguriert ist, funktionieren die Socket-fähigen Planungsoptionen wie vorgesehen: Joboptionen wie `--ntasks-per-socket` und `--sockets-per-node`, und die Socket-basierte Zuweisung mit dem `CR_Socket` oder `CR_Socket_Memory` Werte der Cluster-Ebene `SelectTypeParameters` Einstellung. Weitere Informationen darüber, wie Slurm Sockets, Kerne und Threads zuweist, finden Sie unter Unterstützung für Multi-core/Multi-thread Architekturen in der Slurm-Dokumentation.

AWS PCS konfiguriert keine CPU-Topologie auf Slurm-Versionen vor 25.11, daher kündigen Knoten in diesen Versionen einen Kern pro Socket an und diese Optionen haben keine Auswirkung.

GPU-Finität

Jede GPU in einem Knoten ist einer NUMA-Domäne am nächsten, sodass eine Teilmenge der Kerne des Knotens lokal zu ihr gehört. Eine Aufgabe, die auf Kernen ausgeführt wird, die sich in der Nähe

ihrer GPU befinden, überträgt Daten über einen kürzeren Pfad als eine Aufgabe auf dem anderen Socket. Slurm zeichnet das auf GPU-to-core Affinität pro Gerät (die `Cores` Attribut eines `gres.conf` Datensatz), zusammen mit GPU-to-GPU Verbindungslayout (`Links`).

Auf AWS PCS-Rechenknoten können diese Informationen nicht im Voraus statisch konfiguriert werden, da die Geräteaufzählung von der einzelnen Instanz abhängt und die Geräte erst sichtbar werden, wenn der Knoten gestartet wird. Slurms AutoDetect Der Mechanismus löst das: Wenn ein Knoten bootet, `slurmd` fragt die NVIDIA Management Library (NVML) ab und meldet dem Controller bei der Registrierung die erkannte Affinität.

> **Note**
>
> Während ein Knoten ausgeschaltet ist, weiß Slurm nichts davon GPU-to-CPU Affinität. Die Affinität wird erst verfügbar, wenn der Knoten gestartet und registriert wird. Planungsentscheidungen für ausgeschaltete Knoten werden also ohne Affinität getroffen.

Die automatisch erkannte GPU-Affinität wird in Sockets ausgedrückt. Daher ist eine korrekte CPU-Topologie erforderlich. Wenn ein Knoten eine GPU-Affinität registriert, die nicht zu den konfigurierten Socket-Grenzen des Knotens passt, setzt Slurm den Knoten auf `INVALID_REG` und leert es ab. Weitere Informationen finden Sie unter Beheben Sie Fehler bei ungültigen Knotenregistrierungen (INVALID_REG) in AWS STCK.

Wenn die GPU-Affinität verfügbar ist, kann Slurm Aufgaben auf Kernen platzieren, die sich lokal auf den ihnen zugewiesenen GPUs befinden. Die Generic Resource (GRES) -Affinität wird mit Socket-Granularität behandelt. Die Platzierung hängt von den Joboptionen ab. Zu den Kombinationen, die die erkannte Affinität verwenden, gehören `--ntasks-per-socket` mit `--gpus-per-task`, und `--gpus` mit `--ntasks-per-gpu`.

Topologieverhalten nach Slurm-Version

Was AWS PCS konfiguriert für Sie und was Sie selbst konfigurieren können, hängt von der Slurm-Version des Clusters ab.

Von Slurm 25.11

CPU-Topologie durch benutzerdefinierte Einstellungen

Sie können die CPU-Topologie selbst mit dem konfigurieren `Sockets` und `Parameters` benutzerdefinierte Slurm-Einstellungen für eine Compute-Knotengruppe. Weitere Informationen finden Sie unter Benutzerdefinierte Slurm-Einstellungen für AWS PCS berechnet Knotengruppen.

- Stellen Sie ein `Sockets` auf die Anzahl der Domains, die Slurm als Sockets behandeln soll. AWS PCS berechnet und legt das Matching fest `CoresPerSocket` für dich. Der Wert muss die Anzahl der Kerne des Knotens gleichmäßig teilen.
- Einstellen `Parameters` zu `numa_node_as_socket` oder `l3cache_as_socket` um zu deklarieren, was die Domains sind.

Mit `l3cache_as_socket`, `Sockets` ist erforderlich: AWS PCS kann das L3-Cache-Layout eines Instance-Typs nicht ermitteln, daher geben Sie die Anzahl der L3-Cache-Domänen an. Dies gilt für jede Slurm-Version, einschließlich 26.05 und höher.

Example— L3-Cache-Domains als Sockets auf einem hpc7a.96xlarge Knotengruppe berechnen (24 L3-Cache-Domänen, jeweils 8 Kerne)

```
aws pcs update-compute-node-group \
      --cluster-identifier my-cluster \
      --compute-node-group-identifier my-cng-1 \
      --slurm-configuration \
      'slurmCustomSettings=[{parameterName=Sockets,parameterValue=24},
{parameterName=Parameters,parameterValue=l3cache_as_socket}]'
```

> **Note**
>
> AWS PCS validiert die Struktur eines `Sockets` Wert, nicht dessen Richtigkeit gegenüber der Hardware. Ein Wert, der nicht mit dem tatsächlichen Layout des Knotens übereinstimmt, führt zu einer suboptimalen Platzierung oder zu `INVALID_REG` wenn die automatische GPU-Erkennung aktiv ist.

GPU-Topologie mit AutoDetect (Anmeldung)

Auf Slurm 25.11 ist die automatische GPU-Erkennung standardmäßig ausgeschaltet. Sie melden sich an, indem Sie Folgendes hinzufügen `AutoDetect` zum GPU-Datensatz der Compute-Knotengruppe in `gresCustomSettings`. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter GRES-Einstellungen in AWS 5 STCK.

AutoDetect erfordert die Konfiguration der CPU-Topologie des Knotens. Am 25.11 bedeutet das Einstellung `Sockets` für die Compute-Knotengruppe, wie im vorherigen Abschnitt beschrieben. Ohne sie registrieren Knoten eine GPU-Affinität, die nicht zum Standardlayout mit einem Kern pro Socket passt, und Slurm setzt sie auf `INVALID_REG`.

Von Slurm 26.05

Standard-CPU-Topologie

Von Slurm 26.05, AWS PCS konfiguriert NUMA-based CPU-Topologie standardmäßig. AWS PCS bestimmt die Anzahl der NUMA-Domänen anhand des Instanztyps der Compute-Knotengruppe und legt fest `Sockets`, `CoresPerSocket`, und `Parameters=numa_node_as_socket` auf jedem Knoten.

Die folgenden Regeln gelten:

- Wenn eine Compute-Knotengruppe mehrere Instanztypen verwendet, AWS PCS wendet die Standardtopologie nur an, wenn alle die gleiche Anzahl von NUMA-Domänen haben. Andernfalls AWS PCS lässt die Topologie unkonfiguriert, was dem Verhalten früherer Versionen entspricht.
- Wenn Sie Folgendes festlegen `Sockets` in den benutzerdefinierten Slurm-Einstellungen der Compute-Knotengruppe hat Ihr Wert Vorrang und AWS PCS fügt nichts hinzu `Parameters`. Verwenden Sie dies, um die Standardeinstellung zu überschreiben, z. B. um L3-Cache-Domänen als Sockets mit zu verwenden `Parameters=l3cache_as_socket`, was immer noch deine benötigt `Sockets` Wert.

> **Note**
>
> Die automatische Topologie gilt für Cluster, die mit Slurm 26.05 oder höher erstellt wurden, und für Cluster, wenn Sie sie auf 26.05 oder höher aktualisieren.

Standardmäßig automatische GPU-Erkennung

Von Slurm 26.05, AWS PCS aktiviert standardmäßig die automatische GPU-Erkennung. AWS PCS fügt hinzu `AutoDetect=full` zu dem GPU-Datensatz, den es für eine GPU-Rechenknotengruppe generiert, wenn das Socket-Layout des Knotens bekannt ist, entweder weil AWS PCS hat die CPU-Topologie automatisch konfiguriert, oder weil Sie Folgendes festgelegt haben `Sockets` dich selbst. Wenn das Socket-Layout nicht verfügbar ist, AWS PCS gibt seinen GPU-Datensatz ohne `AutoDetect`, das dem Verhalten früherer Versionen entspricht. Weitere Informationen zur GPU finden Sie unter AWS PCS generiert diesen Datensatz und wie es ihn mit Ihrem eigenen abgleicht, finden Sie unter Wie AWS PCS wendet Ihre gres.conf-Datensätze an.

Konfiguration Multi-Instance GPU (MIG) in AWS STCK

NVIDIA Multi-Instance GPU (MIG) partitioniert eine unterstützte GPU in isolierte GPU-Instanzen, von denen jede über ihre eigenen Speicher- und Rechenressourcen verfügt. Slurm plant jedes MIG-Profil als eigenen GRES-Typ (Generic Resource) ein, sodass sich mehrere Jobs eine physische GPU mit Hardware-Isolierung teilen können.

MIG-Geräte können nicht statisch beschrieben werden in `gres.conf`: Ihre Gerätedateien und UUIDs existieren erst, nachdem die MIG-Instanzen auf dem gestarteten Knoten erstellt wurden. Es gibt also keine `File` oder `Type` Wert, der im Voraus konfiguriert werden soll. Ein AWS PCS konfigurieren Sie MIG mit einem `gres.conf` Datensatz, der nur aktiviert `AutoDetect`, zusammen mit einem `Gres` Einstellung, die die Profile deklariert. `slurmd` erkennt dann die MIG-Geräte, wenn der Knoten gestartet wird. AWS PCS unterstützt `AutoDetect`und daher MIG, auf Slurm-Version 25.11 und höher.

Die Konfiguration von MIG bedeutet, die GPUs auf jedem Knoten zu partitionieren und die resultierenden Profile an Slurm in der Compute-Knotengruppe zu deklarieren. Beides funktioniert nicht alleine, also wenden Sie beide in einer Änderung an. Wenn Slurm nichts über die Partitionen eines Knotens weiß, hat der Knoten kein planbares GRES. Wenn kein Knoten die deklarierten Profile erstellt, werden die Knoten bei der Registrierung geleert. Bei der empfohlenen Aktion für den Knotenlebenszyklus enthält ein Update der Compute-Knotengruppe sowohl das Partitionierungsskript als auch die GRES-Einstellungen.

Voraussetzungen

- Eine Compute-Knotengruppe, die eine verwendet MIG-capable Instanztyp.
- Slurm-Version 25.11 oder höher.

- Konfigurierte CPU-Topologie. Für die automatische GPU-Erkennung ist das Socket-Layout des Knotens erforderlich. Stellen Sie auf Slurm 25.11 ein `Sockets` für die Compute-Knotengruppe; auf Slurm 26.05 und höher AWS PCS konfiguriert es automatisch. Weitere Informationen finden Sie unter Konfiguration der Hardwaretopologie in AWS STCK.

Was ein MIG-Knoten bei jedem Start haben muss

MIG-Modus und MIG-Instances befinden sich auf der GPU, nicht auf der Festplatte, sodass ein benutzerdefiniertes AMI sie nicht übertragen kann. Sie gehen verloren, wenn die Instance gestoppt, ausgesetzt oder beendet wird. Bei jedem Start müssen zuvor alle folgenden Bedingungen erfüllt sein `slurmd` startet:

1. Der MIG-Modus ist aktiviert auf jeder GPU, die MIG-Instanzen bedient. `nvidia-smi -q` meldet eine `Current` und ein `Pending` Modus, und `Current` muss sein `Enabled`.
2. Die MIG-Instanzen existieren, sowohl GPU-Instances als auch Compute-Instances, für jedes Profil in der Compute-Knotengruppe `Gres` Einstellung.
3. Die Profile und Zählungen stimmen überein die `Gres` Einstellung.

Wenn der MIG-Modus aktiviert ist, aber keine MIG-Instanzen existieren, `slurmd` findet die übergeordneten GPUs, aber kein MIG-Gerät. Es protokolliert `MIG mode is enabled, but no MIG devices were found`, dann `Discarding the following config-only GPU due to lack of File specification` für jedes deklarierte Profil. Weil der Knoten weniger GPUs meldet als `Gres` deklariert, lehnt der Controller die Registrierung ab und leert den Knoten mit der Begründung `gres/gpu count reported lower than configured`. Jobs, die eine GPU auf dem Knoten anfordern, schlagen fehl mit `Required node not available`. Wenn Sie die MIG-Instanzen anschließend erstellen, wird der Knoten nicht entlastet; Sie müssen ihn fortsetzen oder ersetzen. Die Aktion im Lebenszyklus des Knotens, beschrieben unter Wenden Sie die Konfiguration mit einer Node-Lifecycle-Aktion an vermeidet dies, indem der Knoten vor der Registrierung ausfällt.

Konfigurieren Sie MIG auf den Instanzen

Die Konfiguration der Instanzen erfordert die folgenden beiden Vorgänge. Führen Sie sie über eine Node-Lifecycle-Aktion aus, bei der die Logik in der Compute-Knotengruppe und nicht in einem AMI, das Sie neu erstellen müssen, beibehalten wird. Siehe Wenden Sie die Konfiguration mit einer Node-Lifecycle-Aktion an. Wenn die GPUs zu diesem Zeitpunkt bereits in einem anderen Prozess gespeichert sind, führen Sie sie früher von einem aus aus `cloud-init` Boothook oder ein `systemd`

Einheit in einem benutzerdefinierten AMI. Siehe Wenn der MIG-Modus weiterhin aussteht. Weitere Hinweise zu den Befehlen finden Sie unter NVIDIA Multi-Instance GPU-Benutzerhandbuch.

1. Aktivieren Sie den MIG-Modus auf den GPUs:

```
nvidia-smi -mig 1
```

   Die Modusänderung kann nicht wirksam werden, solange ein Prozess die GPUs verwendet. Bei der `nodeBootstrapped` Bühne `slurmd` wurde nicht gestartet, sodass auf einem Knoten, auf dem kein anderer GPU-Dienst ausgeführt wird, die Änderung ohne Neustart wirksam wird. Vergewissern Sie sich, dass der angewendete Modus `Enabled` auf jeder GPU mit `nvidia-smi --query-gpu=mig.mode.current --format=csv`.

   > **Important**
   >
   > Wenn der MIG-Modus erhalten bleibt `Pending`, bei einem anderen Prozess sind die GPUs geöffnet. Siehe Wenn der MIG-Modus weiterhin aussteht.

2. Erstellen Sie die GPU-Instanzen und berechnen Sie die Instanzen für die gewünschten Profile. Zum Beispiel, um jede der 8 GPUs in eine zu partitionieren `3g.20gb`, eins `2g.10gb`, und zwei `1g.5gb` Instanzen:

```
for i in 0 1 2 3 4 5 6 7; do
    nvidia-smi mig -i $i -cgi 3g.20gb,2g.10gb,1g.5gb,1g.5gb -C
done
```

   `-cgi` akzeptiert auch numerische Profil-IDs, aber dieselbe ID bedeutet ein anderes Profil auf einem anderen GPU-Modell: `9,14,19,19` ist dieses Layout auf dem 40-GB-A100 in einem `p4d` Instanz, und `3g.40gb`, `2g.20gb`, und zwei `1g.10gb` auf einem 80-GB-Teil. Verwenden Sie die Profilnamen so, dass sie mit den `Gres` Das Einstellen und Erstellen schlägt auf einer GPU fehl, die kein solches Profil hat.

   Erstellen Sie zuerst die größten Profile, wie im Beispiel. MIG-Platzierungen sind festgelegt, sodass ein passendes Layout immer noch nicht erstellt werden kann, wenn die kleinen Profile die Abschnitte einnehmen, die die großen Profile benötigen.

Wenden Sie die Konfiguration mit einer Node-Lifecycle-Aktion an

Konfigurieren Sie die Aktion für den Knotenlebenszyklus wie folgt. Informationen zum Hinzufügen eines Knotens zu einer Compute-Knotengruppe finden Sie unter Konfigurieren Sie Node-Lifecycle-Aktionen in AWS STCK.

Feld	Wert	Warum
Phase des Lebenszyklus	`nodeBoots trapped`	Läuft vorher `slurmd` öffnet die GPUs und registriert den Knoten.
`execution Policy`	`EVERY_BOOT`	Der MIG-Status übersteht einen Stopp, eine Aussetzung oder eine Kündigung nicht.
`arguments`	`["3g.20gb ,2g.10gb, 1g.5gb,1g .5gb"]`	Die MIG-Profile, die auf jeder GPU erstellt werden sollen, namentlich. Ein Skript dient dann zur Berechnung von Knotengruppen mit unterschi edlichen Layouts.
`onError`	`TERMINATE`	Ersetzt einen Knoten, der nicht partitioniert werden kann, anstatt ihn ohne seine MIG-Geräte registrie ren zu lassen. Verwenden Sie beim Debuggen `STOP_SEQUENCE` um die Instanz zur Überprüfung aufzubewahren.

Ein Skript, das bei jedem Start ausgeführt wird, muss idempotent sein. Die folgende Gliederung verwendet das Layout pro GPU als erstes Argument und behandelt jeweils eine GPU. Es überspringt eine GPU, die bereits über dieses Layout verfügt, partitioniert jede andere neu und wird ungleich Null beendet, wenn sie nicht fertig werden kann, wodurch der Knoten fehlschlägt. Der Knoten wird nie neu gestartet. Informationen dazu, wie Argumente ein Skript erreichen, finden Sie unter Übergabe von Argumenten an Skripts.

```
#!/usr/bin/env bash
# Partition every GPU as $1 specifies, then verify the result. Expected
# nvidia-smi failures are handled inline, so this doesn't use 'set -e'.
set -uo pipefail

# $1 - per-GPU MIG profile names, comma-separated, largest first, as in Gres.
```

```
PROFILES="${1:-}"
if ! echo "$PROFILES" | grep -Eq '^[0-9]+g\.[0-9]+gb(,[0-9]+g\.[0-9]+gb)*$'; then
    echo "ERROR: pass the per-GPU MIG profiles as the first argument," >&2
    echo "       for example 3g.20gb,2g.10gb,1g.5gb,1g.5gb" >&2
    exit 1
fi
EXPECTED=$(echo "$PROFILES" | tr ',' '\n' | sort)

# Fail, rather than exit 0, when there is no GPU to configure.
GPUS=$(nvidia-smi --query-gpu=index --format=csv,noheader) || {
    echo "ERROR: nvidia-smi failed. This node has no usable NVIDIA driver." >&2
    exit 1
}
if [ -z "$GPUS" ]; then
    echo "ERROR: no NVIDIA GPUs detected." >&2
    exit 1
fi

# The MIG profiles on GPU $1, in the same form as EXPECTED. nvidia-smi -L lists
# what NVML reports, which is what slurmd autodetects.
mig_devices() {
    nvidia-smi -L | awk -v gpu="$1" '
        /^GPU [0-9]+:/ { current = ($2 + 0); next }
        $1 == "MIG" && current == gpu { print $2 }
    ' | sort
}

for i in $GPUS; do
    # 1. Skip a GPU that already has the requested layout.
    if [ "$(mig_devices "$i")" = "$EXPECTED" ]; then
        echo "GPU $i: already configured as $PROFILES"
        continue
    fi

    # 2. Enable MIG mode. Ignore the exit status; step 3 checks the result.
    enable_output=$(nvidia-smi -i "$i" -mig 1 2>&1) || true

    # 3. Fail if the mode isn't applied. Don't reboot: that would interrupt
    #    the AWS PCS bootstrap.
    current=$(nvidia-smi -i "$i" --query-gpu=mig.mode.current --format=csv,noheader)
    pending=$(nvidia-smi -i "$i" --query-gpu=mig.mode.pending --format=csv,noheader)
    if [ "$current" != "Enabled" ]; then
        echo "ERROR: GPU $i MIG mode is '$current', pending '$pending'." >&2
        echo "$enable_output" >&2
```

```
        if [ "$pending" = "Enabled" ]; then
            echo "The enable is pending, so a process is holding the GPU. Stop it," >&2
            echo "or configure MIG mode in a boothook instead." >&2
        elif [ "$current" = "[N/A]" ] || [ "$current" = "N/A" ]; then
            echo "This GPU does not support MIG." >&2
        fi
        exit 1
    fi

    # 4. Recreate the layout from scratch, because instances can't be added to
    #    a partial one. Nothing uses them yet, because slurmd hasn't started.
    nvidia-smi mig -i "$i" -dci >/dev/null 2>&1 || true
    nvidia-smi mig -i "$i" -dgi >/dev/null 2>&1 || true
    if ! nvidia-smi mig -i "$i" -cgi "$PROFILES" -C; then
        echo "ERROR: MIG instance creation failed on GPU $i" >&2
        exit 1
    fi

    # 5. Check which profiles exist, not only how many, so a layout that
    #    doesn't match Gres fails here instead of at job time.
    actual=$(mig_devices "$i")
    if [ "$actual" != "$EXPECTED" ]; then
        echo "ERROR: GPU $i has the wrong MIG layout." >&2
        echo "  expected: $(echo "$EXPECTED" | tr '\n' ' ')" >&2
        echo "  found:    $(echo "$actual" | tr '\n' ' ')" >&2
        exit 1
    fi
    echo "GPU $i: configured as $PROFILES"
done
```

Informationen zum Festlegen des Skripts und der GRES-Einstellungen in einem Aufruf finden Sie im Beispiel unter Deklarieren Sie die MIG-Profile für Slurm.

Wenn Sie das Skript anpassen, behalten Sie beide Schecks bei. Ohne die Modusüberprüfung ein Knoten, dessen MIG-Modus nie `Enabled` fährt fort, sich ohne seine MIG-Geräte zu registrieren, und die Ursache zeigt sich nur in der `slurmd` protokollieren. Ohne die Layoutüberprüfung registriert sich ein Knoten, der die falschen Profile erstellt, normal und schlägt nur bei den Jobs fehl, die die deklarierten Profile anfordern. Informationen darüber, wo die Ausgabe des Skripts geschrieben wird, finden Sie unter Loggen und Debuggen.

Wenn der MIG-Modus weiterhin aussteht

Die Aktivierung des MIG-Modus schlägt fehl, solange bei einem Prozess die GPUs geöffnet sind. `nvidia-smi` meldet die Aktivierung als ausstehend, zum Beispiel mit `In use by another client`, und es wird erst wirksam, wenn diese Prozesse beendet werden oder der Knoten neu gestartet wird. Auf einem AWS PCS-Rechenknoten, bei dem Prozess in der Regel ein GPU-Monitoring-Agent wie NVIDIA DCGM oder ein Agent, der im AMI enthalten ist, oder `slurmd` selbst, wenn MIG danach konfiguriert wird `slurmd` startet. Sie können den Computer nicht von der Lifecycle-Aktion aus neu starten. Verwenden Sie daher eine der folgenden Fixes:

- Stoppen Sie den Prozess, der die GPUs enthält. Deaktivieren oder entfernen Sie im benutzerdefinierten AMI alle Daemons, die die GPUs öffnen, bevor die MIG-Konfiguration ausgeführt wird. Entfernen Sie auch alle Daemons, die MIG selbst konfiguriert. Zwei Komponenten, die dieselben GPUs partitionieren, erzeugen ein Layout, das nicht dem entspricht `Gres` Einstellung.
- Konfigurieren Sie MIG in einem Boothook. Wenn Sie den Vorgang nicht stoppen können, verschieben Sie den MIG-Moduswechsel und die Profilerstellung in eine `cloud-init` Boothook in den Benutzerdaten der Startvorlage. Ein Boothook läuft vor dem AWS PCS-Bootstrap und vor dem Start der AMI-Dienste, sodass der Moduswechsel ohne Neustart wirksam wird. Wie man einen schreibt, finden Sie unter Beispiel: Führen Sie frühe Startvorgänge mit einem Cloud-Init-Boothook aus. Behalten Sie die Lifecycle-Aktion im Auge: Bei jedem Start wird das Layout bestätigt und der Knoten schlägt fehl, wenn der Boothook es nicht angewendet hat.

Deklarieren Sie die MIG-Profile für Slurm

Deklarieren Sie beide GRES-Einstellungen für die Compute-Knotengruppe:

- In `gresCustomSettings`, ein GPU-Datensatz, der nur aktiviert `AutoDetect`. Nicht setzen `Name`, `Type`, oder `File`. Die MIG-Geräte werden auf dem Knoten erkannt.
- In `slurmCustomSettings`, ein `Gres` Einstellung, die die MIG-Profile und die Anzahl der MIG-Profile deklariert, die Jobs anfordern können. Die Profilnamen und die Anzahl müssen mit den auf dem Knoten erstellten MIG-Instanzen übereinstimmen.

Example— Anwenden beider Teile in einem Update (8 GPUs, jeweils in eine partitioniert) 3 g, 20 GB, eins 2 g, 10 GB, und zwei 1 g, 5 GB)

```
aws pcs update-compute-node-group \
```

```
    --cluster-identifier my-cluster \
    --compute-node-group-identifier my-cng-1 \
    --slurm-configuration '{
      "gresCustomSettings": [
        { "AutoDetect": "nvml" }
      ],
      "slurmCustomSettings": [
        { "parameterName": "Gres", "parameterValue":
 "gpu:3g.20gb:8,gpu:2g.10gb:8,gpu:1g.5gb:16" }
      ]
    }' \
    --node-lifecycle-actions '{
      "stages": {
        "nodeBootstrapped": [
          {
            "name": "configure-mig",
            "scriptSource": {
              "scriptLocation": "s3://my-bucket/configure-mig.sh"
            },
            "arguments": ["3g.20gb,2g.10gb,1g.5gb,1g.5gb"],
            "executionPolicy": "EVERY_BOOT",
            "onError": "TERMINATE"
          }
        ]
      }
    }'
```

Das Beispiel übergibt beide Parameter als JSON, weil `Gres` Der Wert enthält Kommas, für die CLI-Kurzsyntax ein Escape-Symbol erforderlich wäre.

Behalten `arguments` und `Gres` konsistent: `3g.20gb,2g.10gb,1g.5gb,1g.5gb` auf jeder der 8 GPUs ergeben die 8, 8 und 16 Instanzen `gpu:3g.20gb:8,gpu:2g.10gb:8,gpu:1g.5gb:16` deklariert.

Die Einschränkungen zwischen den beiden Einstellungen finden Sie unter Einschränkungen zwischen gres.conf und der Gres-Einstellung. Weil AWS PCS kann Geräte, die beim Booten erkannt werden, nicht zählen. Es akzeptiert alle Profilnamen und zählt in `Gres` Einstellung für einen Datensatz, der nur aktiviert `AutoDetect`. Die Konsistenz mit den tatsächlichen MIG-Instanzen wird nur überprüft, wenn sich der Knoten registriert.

Wenn ein Knoten bootet und sich registriert, kündigt er beispielsweise die erkannten Profile mit ihrer Socket-Affinität an `Gres=gpu:3g.20gb:8(S:0),gpu:1g.5gb:28(S:1)`. Ein Job, der

ein Profil anfordert, wie `--gres=gpu:3g.20gb:1`, wird auf einer einzelnen MIG-Instanz platziert und erhält ihre UUID `CUDA_VISIBLE_DEVICES` (zum Beispiel `MIG-e7aa6185-06d5-53e9-a4d9-00f7670f741e`), sodass CUDA den Job auf diese MIG-Instanz beschränkt.

> **Note**
>
> Der Controller protokolliert Warnungen wie `Ignoring file-less GPU gpu:3g.20gb from final GRES list` für MIG-Aufzeichnungen. Das wird erwartet: Das autoritative GRES stammt aus der Knotenregistrierung.

Einschränkungen

Da eine MIG-Konfiguration keine Geräte benennt, kennt der Controller das tatsächliche GRES eines Knotens erst, nachdem der Knoten gestartet und registriert wurde. Bis sich ein Knoten einmal registriert hat, speichert der Controller nur die Profilzählungen von `Gres` Einstellung. Dies führt zu der folgenden Einschränkung. Sie gilt für jeden GPU-Datensatz, der nur aktiviert `AutoDetect`und daher für jede MIG-Rechenknotengruppe.

> **Note**
>
> Verwenden Sie als bewährte Methode eine statische Rechenknotengruppe für MIG: set `minInstanceCount` entspricht `maxInstanceCount` in der Skalierungskonfiguration. Die Knoten registrieren sich dann, wenn sie gestartet werden, bevor ihnen ein Job zugewiesen wird, wodurch die Einschränkung umgangen wird.

Jobs, die zugewiesen werden, bevor sich ein Knoten zum ersten Mal registriert, können ohne GPU gestartet werden

Einem Auftrag, der einem Knoten zugewiesen ist, der ausgeschaltet ist und sich noch nie registriert hat, beispielsweise beim ersten Start einer neu erstellten Rechenknotengruppe, wird eine Profilanzahl zugewiesen, aber kein bestimmtes Gerät, da die Geräte noch nicht bekannt sind. Wenn der Knoten dann bootet und der Job gestartet wird, ist dem Job kein MIG-Gerät zugewiesen: `CUDA_VISIBLE_DEVICES` ist nicht gesetzt, und weil AWS PCS schränkt den Gerätezugriff nicht über Cgroups ein, der Job kann jedes MIG-Gerät auf dem Knoten verwenden. Slurm meldet den Job als `RUNNING` allerdings, und die MIG-Instanz, für die der Job gezählt wurde, wird dem nächsten Job zugewiesen, der dieses Profil anfordert, sodass beide Jobs auf derselben MIG-Instanz landen

können. Jobs, die dem Knoten zugewiesen werden, nachdem er sich einmal registriert hat, auch nach späteren Aus- und Einschaltzyklen, erhalten erwartungsgemäß ihr Gerät.

Um diesen Zustand zu erkennen, überprüfen Sie zu Beginn des Jobs, dass `CUDA_VISIBLE_DEVICES` ist gesetzt, und beenden Sie den Job oder stellen Sie ihn erneut in die Warteschlange, wenn dies nicht der Fall ist:

```
if [ -z "${CUDA_VISIBLE_DEVICES-}" ]; then
    scontrol requeue "$SLURM_JOB_ID"
    exit 1
fi
```

Ein Auftrag, der erneut in die Warteschlange gestellt wurde, kann nach dem folgenden Befehl erneut ausgeführt werden `requeue_delay` Scheduler-Parameter, standardmäßig 5 Sekunden. Bis dahin hat sich der Knoten registriert, sodass der Job beim nächsten Start ein Gerät erhält.

Erweitern Sie die Slurm-Funktionalität auf AWS PCS mit SPANK-Plugins

Verwenden Sie SPANK (Slurm) Plug-in Plugins (Architecture for Node und Job Kontrol), um das Verhalten von Slurm beim Jobstart und bei der Ausführung von Jobs zu erweitern und zu modifizieren AWS PCS-Cluster. SPANK-Plugins bieten eine generische Schnittstelle zum Abfangen und Ändern von Phasen des Jobstarts.

Installieren Sie SPANK-Plugins auf Ihrem Compute-Node-AMI und konfigurieren Sie sie, um das Verhalten Ihres Slurm-Clusters an Ihre Workload-Anforderungen anzupassen. Weitere Informationen zu SPANK finden Sie im SPANK-Dokumentation auf der SchedMD-Website.

Inhalt

- Installieren Sie die SPANK-Plugins auf AWS STCK
- Konfiguriere die SPANK-Plugins auf AWS STCK
- Häufig gestellte Fragen zu SPANK-Plugins auf AWS STCK

Installieren Sie die SPANK-Plugins auf AWS STCK

Folgen Sie der Dokumentation des Plugins, um SPANK-Plugins auf Ihrem AMI zu installieren.

Kompilieren Sie SPANK-Plugins für die spezifische Slurm-Version auf Ihrem Cluster. Das Slurm-Installationsprogramm, bereitgestellt von AWS PCS speichert Slurm in `/opt/aws/pcs/scheduler/slurm-version`. Wenn Sie das Plugin kompilieren, geben Sie die Slurm-Version an.

Das folgende Beispiel zeigt, wie die Slurm-Version für einige Plugins angegeben wird:

```
export CFLAGS="-I/opt/aws/pcs/scheduler/slurm-version/include"
```

Wenn Sie mehrere Slurm-Versionen im AMI haben, kompilieren Sie das Plugin für jede Version. Speichern Sie die kompilierten Plugins in versionierten Ordnern.

Das folgende Beispiel zeigt, wie der Zielordner für einige Plugins angegeben wird:

```
export DESTDIR="your-preferred-versioned-path"
```

> **Important**
>
> Plugins benötigen möglicherweise andere Variablen. Sieh dir die offizielle Dokumentation für das Plugin an, das du installierst.

Konfiguriere die SPANK-Plugins auf AWS STCK

Speichern Sie Konfigurationsdateien standardmäßig in `/etc/aws/pcs/scheduler/slurm-version/plugstack.conf.d/`.

Um Ihre SPANK-Konfiguration an einem anderen Ort zu speichern, fügen Sie Ihre Speicherorte zu einer Konfigurationsdatei im Standardverzeichnis hinzu.

Das folgende Beispiel zeigt, wie Sie Konfigurationsdateien aus anderen Verzeichnissen einbeziehen:

```
# content of /etc/aws/pcs/scheduler/slurm-version/any-filename.conf
include path-to-your-configuration-folder/*.conf
include path-to-a-second-configuration-folder/*.conf
```

Speichern Sie jede Konfiguration in einer speziellen Datei oder in einer gemeinsamen Datei. Sie können mehrere Konfigurationsdateien verwenden.

Die folgenden Beispiele zeigen Beispielkonfigurationsdateien:

```
# content of path-to-your-or-default-config-folder/filename-1.conf
required path-to-plugin-1 arguments
optional path-to-plugin-2 arguments
```

```
# content of path-to-your-or-default-config-folder/filename-2.conf
required path-to-plugin-3 arguments
```

Weitere Informationen zur Konfiguration Ihrer Plugins finden Sie in der Dokumentation zur SPANK-Konfiguration auf der SchedMD-Website.

Important

Legen Sie Ordnerberechtigungen fest, um unbefugte Änderungen an Ihrer Plugin-Konfiguration zu verhindern.

Note

AWS PCS verwaltet Ihre SPANK-Plugins nicht. Wenn Sie Fehler im Zusammenhang mit Plugins erhalten, überprüfen Sie die Fehlerprotokolle auf Ihren Rechenknoten.

Note

Slurm protokolliert beim Laden Ihrer SPANK-Konfiguration fälschlicherweise einen Fehler ähnlich dem folgenden:

```
error: "Include" failed in file /etc/slurm/plugstack.conf line 3
```

Sie können diesen Fehler ignorieren. Es hat keinen Einfluss darauf, wie SPANK-Plugins funktionieren.

Häufig gestellte Fragen zu SPANK-Plugins auf AWS STCK

In diesem Abschnitt werden häufig gestellte Fragen zur Installation und Konfiguration von SPANK-Plugins auf behandelt AWS PCS-Cluster.

Muss ich SPANK-Plugins sowohl auf Anmelde- als auch auf Rechenknoten installieren?

Einige SPANK-Plugins müssen nicht auf allen Knoten installiert werden. Für eine bessere Kompatibilität empfehlen wir jedoch, alle SPANK-Plugins auf jedem Knoten zu installieren.

Welche zusätzliche Konfiguration ist für den produktiven Einsatz von SPANK-Plugins erforderlich?

Neben der in den Beispielen gezeigten grundlegenden Installation und Konfiguration ist für Produktionsbereitstellungen in der Regel eine zusätzliche Einrichtung erforderlich. Container-based Für Plugins wie Pyxis müssen Sie möglicherweise Umgebungsvariablen für Enroot festlegen, PMI (Process Management Interface) aktivieren und Berechtigungen für die Container-Laufzeit konfigurieren. In der Dokumentation des jeweiligen Plugins finden Sie detaillierte Anforderungen an die Produktionsbereitstellung.

Wie behebe ich Probleme mit dem SPANK-Plugin?

AWS PCS verwaltet keine SPANK-Plugins. Untersuchen Sie die Fehlerprotokolle auf Ihren Rechenknoten, um Probleme zu beheben.

Verwenden Sie die Slurm CLI Filter Plugins, um die Auftragsübermittlung in anzupassen AWS 5 STCK

AWS PCS unterstützt Slurm CLI-Filter-Plugins zur Ausführung benutzerdefinierter Lua-Skripte, die die Parameter für die Auftragsübermittlung auf Anmelde- und Rechenknoten validieren und ändern. Ausführliche Informationen zu CLI-Filter-Plugins finden Sie im API-Dokumentation für das cli_filter-Plug-In auf der SchedMD-Website.

Voraussetzungen

CLI-Filter-Plugins benötigen Slurm-Version 24.11 oder höher und ein Lua-Skript, das auf allen Anmelde- und Rechenknoten bereitgestellt wird.

> **Important**
>
> Für die Slurm-Versionen 24.11 und 25.05 erfordern die CLI-Filter-Plugins die Installation von Slurm mit AWS PCS Slurm-Installationsprogramm (Version 24.11.6-2+ oder 25.05.4-1+). Weitere Informationen zur Installation von Slurm finden Sie unter Schritt 3 — Slurm installieren.

Einschränkungen und Sicherheitsüberlegungen

- Durchsetzung der Sicherheitsvorkehrungen — CLI-Filter-Plugins können von jedem Benutzer problemlos umgangen werden und dürfen nicht für sicherheitskritische Richtlinien verwendet werden. Benutzer können CLI-Filter-Plugins deaktivieren, indem sie eine benutzerdefinierte Konfiguration bereitstellen, die `CLIFilterPlugins` beim Senden von Aufträgen deaktiviert.
- Nur Lua-Implementierung — Die Lua-Skript-Implementierung wird unterstützt. Die C-Implementierung wird nicht unterstützt.

Themen

- Konfigurieren Sie die Slurm CLI Filter Plugins auf einem AWS PCS-Cluster
- Verwenden Sie Amazon S3, um ein CLI-Filter-Plugin-Skript bereitzustellen in AWS 5 STCK
- Übersetzen Sie ein Slurm Job Submit-Plugin-Skript, um das CLI-Filter-Plugin in zu verwenden AWS 5 STCK
- Häufig gestellte Fragen zu Slurm CLI Filter Plugins in AWS 5 STCK
- Behebung von Problemen mit dem Slurm CLI Filter Plugin in AWS 5 STCK

Konfigurieren Sie die Slurm CLI Filter Plugins auf einem AWS PCS-Cluster

Konfigurieren Sie CLI-Filter-Plugins, wenn Sie ein neues erstellen AWS PCS-Cluster. Sie können CLI-Filter-Plugins auf vorhandenen Clustern mithilfe der Update API oder der Konsole aktivieren oder deaktivieren, ohne den Cluster neu zu erstellen.

Voraussetzungen

Bevor Sie die CLI-Filter-Plugins konfigurieren, führen Sie die folgenden Aufgaben aus:

- Stellen Sie sicher, dass Sie Slurm Version 24.11 oder höher verwenden. Für 24.11 und 25.05 siehe the section called “Voraussetzungen”.
- Schreiben und testen Sie ein Lua-Skript, das die CLI Filter Plugin API implementiert
- Wählen Sie eine Methode, um Ihr Skript auf allen Cluster-Instances (AMI, S3 oder Dateisystem) bereitzustellen

Aktivieren Sie die CLI-Filter-Plugins auf einem neuen Cluster

AWS PCS console

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie Cluster erstellen.
4. Unter Scheduler-Einstellungen, erweitern Zusätzliche Scheduler-Einstellungen.
5. Füge eine neue benutzerdefinierte Slurm-Einstellung hinzu mit Name des Parameters gesetzt auf `CliFilterPlugins` und Parameterwert gesetzt auf `cli_filter/lua`.
6. Vervollständigen Sie die restliche Cluster-Konfiguration und wählen Sie Cluster erstellen.

AWS PCS API

Stellen Sie das zur Verfügung `slurmCustomSettings` Konfiguration in Ihrem Anruf an die `CreateCluster` API-Aktion. Stellen Sie das ein `parameterName` zu `CliFilterPlugins` und `parameterValue` zu `cli_filter/lua`. Weitere Informationen finden Sie unter CreateCluster in der AWS PCS-API-Referenz.

Das folgende Beispiel verwendet AWS CLI um den aufzurufen `CreateCluster` API-Aktion. Die benutzerdefinierte Einstellung `CliFilterPlugins=cli_filter/lua` aktiviert CLI-Filter-Plugins.

```
aws pcs create-cluster --cluster-name cluster-name \
--scheduler type=SLURM,version=25.11 \
--size SMALL \
--networking subnetIds=cluster-subnet-id,securityGroupIds=cluster-security-group-id
 \
--slurm-configuration \
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/
lua"}]'
```

Stellen Sie CLI-Filter-Plugin-Skripte bereit

So stellen Sie CLI-Filter-Plugin-Skripte in Ihrem Cluster bereit

1. Stellen Sie sicher, dass auf allen AMIs, die in Compute Node-Gruppen verwendet werden, Slurm über das installiert ist AWS PCS Slurm-Installationsprogramm.

> **Note**
>
> Wenn Sie das verwenden AWS PCS-Beispiel-AMI für alle Rechenknotengruppen, überspringen Sie diesen Schritt. Slurm ist bereits installiert.

2. Stellen Sie Ihr Lua-Skript bereit `/etc/aws/pcs/scheduler/slurm-version/cli_filter.lua` auf allen Instanzen im Cluster.

   Zum Beispiel für Slurm Version 25.11:

   ```
   /etc/aws/pcs/scheduler/slurm-25.11/cli_filter.lua
   ```

   > **Note**
   >
   > Ab der Slurm-Version 25.11 können Sie den Pfad durch anpassen `cli_filter_lua_path` in der `CliFilterParameters` Benutzerdefinierte Slurm-Einstellung. Dieser Parameter lautet standardmäßig `/etc/aws/pcs/scheduler/slurm/cli_filter.lua`. Die AWS Der PCS-Agent verknüpft diesen Pfad ab v1.4.1-1 mit dem versionierten Speicherort.

3. Starten Sie alle Anmelde- und Rechenknoten mit Ihren vorbereiteten AMIs.
4. Testen Sie die Auftragsübermittlung, um sicherzustellen, dass das CLI-Filter-Plugin korrekt ausgeführt wird.

Aktivieren oder deaktivieren Sie die CLI-Filter-Plugins auf vorhandenen Clustern

Sie können CLI-Filter-Plugins auf vorhandenen Clustern aktivieren oder deaktivieren, ohne Ihre Infrastruktur neu aufzubauen. Weitere Informationen finden Sie unter Aktualisierung eines Clusters in AWS STK..

AWS PCS console

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, der aktualisiert werden soll.
4. Wählen Sie Bearbeiten Aktion.

5. Auf der Seite Cluster bearbeiten unter Zusätzliche Scheduler-Einstellungen:
   - Um die CLI-Filter-Plugins zu aktivieren: Füge eine neue benutzerdefinierte Slurm-Einstellung hinzu mit Name des Parameters gesetzt auf `CliFilterPlugins` und Parameterwert gesetzt auf `cli_filter/lua`.
   - Um die CLI-Filter-Plugins zu deaktivieren: Entfernen Sie die vorhandenen `CliFilterPlugins` Einstellung.
6. Wählen Sie Cluster aktualisieren um die Änderungen einzureichen.
7. Überwachen Sie den Cluster-Status, der während des Vorgangs als „Aktualisierung“ und nach Abschluss der Aktualisierung als „Aktiv“ angezeigt wird.

AWS PCS API

Verwenden Sie das `UpdateCluster` API-Aktion, um CLI-Filter-Plugins zu aktivieren oder zu deaktivieren. Weitere Informationen finden Sie unter UpdateCluster in der AWS PCS-API-Referenz.

So aktivieren Sie CLI-Filter-Plugins auf einem vorhandenen Cluster:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'slurmCustomSettings=[{parameterName=CliFilterPlugins,parameterValue="cli_filter/
lua"}]'
```

So deaktivieren Sie die CLI-Filter-Plugins auf einem vorhandenen Cluster:

```
aws pcs update-cluster --cluster-identifier my-cluster \
--slurm-configuration \
'slurmCustomSettings=[]'
```

Erwartete Ergebnisse

Nachdem Sie die Konfiguration abgeschlossen haben:

- Ihr Cluster wird mit aktiviertem CLI-Filter-Plugin erstellt
- Auftragsübermittlungen lösen Ihre benutzerdefinierte Validierungslogik aus, bevor sie den Slurm-Controller erreichen

- Non-compliant Jobs werden mit Ihren benutzerdefinierten Fehlermeldungen abgelehnt
- Konforme Jobs werden normal durch den Slurm-Scheduler ausgeführt

Fehlerbehebung

Das CLI-Filter-Plugin-Skript fehlt auf einem Knoten

Symptome: Die Auftragsübermittlung schlägt sofort mit einem Fehler beim Laden des Plugins fehl.

Wahrscheinliche Ursache: Das Skript wurde nicht auf allen Instanzen bereitgestellt oder der Dateipfad oder -name ist falsch.

Lösung: Stellen Sie sicher, dass das Skript auf allen Anmelde- und Rechenknoten im richtigen Pfad mit dem exakten Dateinamen vorhanden ist `cli_filter.lua`.

Ungültige Konfiguration des CLI-Filter-Plug-ins

Symptome: Die Clustererstellung schlägt mit einem Validierungsfehler fehl.

Wahrscheinliche Ursache: `CliFilterPlugins` Parameter nicht gesetzt auf `cli_filter/lua` formatieren.

Auflösung: Verwenden Sie den exakten Parameterwert `cli_filter/lua` in `slurmCustomSettings`.

Verwenden Sie Amazon S3, um ein CLI-Filter-Plugin-Skript bereitzustellen in AWS 5 STCK

Verwenden Sie S3, um Ihr CLI-Filter-Plugin-Skript bereitzustellen, wenn Sie die Logik für die Auftragsübermittlung in einem Live-Cluster aktualisieren möchten, ohne AMIs neu erstellen zu müssen. Bei diesem Ansatz wird das Skript während des Instance-Starts mithilfe von Benutzerdaten von S3 heruntergeladen.

Voraussetzungen

Bevor Sie Ihr Skript mit S3 bereitstellen, führen Sie die folgenden Aufgaben aus:

- Stellen Sie sicher, dass Sie Slurm Version 24.11 oder höher verwenden. Für 24.11 und 25.05 siehe the section called “Voraussetzungen”.

- Erstellen Sie einen S3-Bucket mit Ihrem CLI-Filter-Plug-in-Lua-Skript
- Konfigurieren Sie das IAM-Instanzprofil mit Lesezugriff auf den S3-Bucket
- Richten Sie den S3 VPC Gateway-Endpunkt für den direkten Zugriff ohne Internet ein
- Bereiten Sie das Benutzerdatenskript für den Download von S3 vor

So stellen Sie das CLI-Filter-Plugin-Skript mit S3 bereit

1. Laden Sie Ihre hoch `cli_filter.lua` Skript in Ihren S3-Bucket.
2. Konfigurieren Sie Ihr IAM-Instanzprofil mit S3-Leseberechtigungen für den Bucket.
3. Fügen Sie Shell-Code zu den Benutzerdaten Ihrer Startvorlage hinzu, um das Skript herunterzuladen:

   ```
   aws s3 cp s3://my-bucket/cli_filter.lua /etc/aws/pcs/scheduler/slurm-25.11/
   cli_filter.lua
   chmod 644 /etc/aws/pcs/scheduler/slurm-25.11/cli_filter.lua
   ```
4. Stellen Sie Compute Node-Gruppen mit Ihren aktualisierten Startvorlagen bereit.
5. Testen Sie die Auftragsübermittlung, um die Skriptfunktionalität zu überprüfen.

Erwartete Ergebnisse

Nachdem Sie die S3-Bereitstellung abgeschlossen haben:

- Das CLI-Filter-Plugin-Skript wird beim Start automatisch auf alle Instanzen heruntergeladen
- Skriptaktualisierungen in S3 werden auf neu gestarteten Instances wiedergegeben
- Die Richtlinien für die Auftragsübermittlung werden im gesamten Cluster einheitlich durchgesetzt

Fehlerbehebung

S3-Zugriff verweigert

Symptome: Der Instanzstart schlägt fehl oder das Skript wurde nicht heruntergeladen.

Wahrscheinliche Ursache: Fehlende IAM-Berechtigungen oder S3-VPC-Endpunkt.

Lösung: Stellen Sie sicher, dass das IAM-Instanzprofil `s3:GetObject` Die Berechtigung und der S3-VPC-Endpunkt sind konfiguriert.

Übersetzen Sie ein Slurm Job Submit-Plugin-Skript, um das CLI-Filter-Plugin in zu verwenden AWS 5 STCK

Übersetzen Sie Ihr vorhandenes Lua-Skript für das Job Submit Plugin in das CLI-Filter-Plugin, wenn Sie von anderen Slurm-Umgebungen migrieren. Der Übersetzungsprozess beinhaltet die Aktualisierung von Funktionsnamen und Feldzugriffsmustern, damit sie mit der CLI Filter Plugin-API funktionieren.

Voraussetzungen

Bevor Sie Ihr Skript übersetzen, führen Sie die folgenden Aufgaben aus:

- Überprüfe dein vorhandenes Lua-Skript für das Job Submit Plugin
- Machen Sie sich mit den Unterschiede zwischen den Job Submit- und CLI-Filter-Plugin-APIs vertraut
- Greifen Sie auf die Dokumentation zum Slurm CLI Filter Plugin zu

Um das Job Submit Plugin-Skript in das CLI-Filter-Plugin zu übersetzen

1. Überprüfen Sie Ihre vorhandenen Skriptfunktionen für das Job Submit Plugin (`slurm_job_submit`, `slurm_job_modify`).
2. Identifizieren Sie äquivalente Funktionen des CLI-Filter-Plug-ins:
   - `slurm_job_submit` wird `slurm_cli_pre_submit`
   - Hinzufügen `slurm_cli_setup_defaults` für die Standardparametereinstellung
   - Hinzufügen `slurm_cli_post_submit` für Aktionen nach der Einreichung
3. Übersetzen Sie die Logik zur Jobvalidierung von `job_desc` Felder nach `options` Array-Zugriff:
   - `job_desc.account` wird `options["account"]`
   - `job_desc.partition` wird `options["partition"]`
   - `job_desc.features` wird `options["constraint"]`
4. Aktualisieren Sie die Protokollierung von Anrufen `slurm.log_user()` zu `slurm.log_error()`.
5. Testen Sie Ihr übersetztes Skript auf einem Entwicklungscluster.
6. Führen Sie die Bereitstellung auf Ihrem Produktionscluster gemäß dem standardmäßigen Bereitstellungsprozess für das CLI-Filter-Plugin durch.

Erwartete Ergebnisse

Nachdem Sie die Übersetzung abgeschlossen haben:

- Ihr übersetztes Skript bietet eine gleichwertige Validierung der Auftragseingabe
- Benutzern werden ähnliche Fehlermeldungen und Eingabeaufforderungen wie bei Ihrem ursprünglichen Job Submit Plugin angezeigt
- Die Richtlinien für die Einreichung von Jobs werden während der Migration zu beibehalten AWS 5 STÜCK

Fehlerbehebung

Fehler bei der Skript-Übersetzung

Symptome: Die Auftragsübermittlung schlägt mit Fehlern bei der Lua-Ausführung fehl.

Wahrscheinliche Ursache: Falscher Feldzugriff oder Funktionsaufrufe im übersetzten Skript.

Auflösung: Lesen Sie die API-Dokumentation des CLI Filter Plug-ins und vergleichen Sie die Feldzuordnungen zwischen den Schnittstellen Job Submit und CLI Filter.

Häufig gestellte Fragen zu Slurm CLI Filter Plugins in AWS 5 STCK

Lesen Sie diese häufig gestellten Fragen zu CLI-Filter-Plugins.

Was ist der Unterschied zwischen dem CLI-Filter-Plugin und dem Job Submit-Plugin?

Das CLI-Filter-Plugin wird clientseitig auf Anmelde- und Rechenknoten ausgeführt, bevor die Auftragsübermittlung den Controller erreicht, während das Job Submit Plugin nach der Auftragsübermittlung serverseitig auf dem Controller ausgeführt wird. Das CLI-Filter-Plugin kann von Benutzern umgangen werden, hält aber keine Controller-Sperren. Job Submit ist zwar sicher, kann aber die Cluster-Leistung während der Ausführung beeinträchtigen.

Tut AWS PCS unterstützt das Slurm Job Submit Plugin?

Nein, das Job Submit Plugin wird nicht unterstützt in AWS STCK. Verwenden Sie stattdessen das CLI Filter Plugin für die Validierung und Änderung der Auftragsübermittlung.

Kann ich das CLI-Filter-Plugin zur Durchsetzung der Sicherheitsvorkehrungen verwenden?

Nein, das CLI-Filter-Plugin kann von bestimmten Benutzern umgangen werden und sollte nicht zur Durchsetzung der Sicherheitsvorkehrungen verwendet werden. Verwenden Sie es eher für Verbesserungen der Benutzererfahrung, für die Einstellung von Standardparametern und für Richtlinien als für sicherheitskritische Richtlinien.

Warum muss sich das Skript auf allen Rechenknoten befinden, nicht nur auf Anmeldeknoten?

Slurm-Befehle wie `srun` können innerhalb von Job-Skripten auf Rechenknoten ausgeführt werden, was auch die Ausführung des CLI Filter Plugins auslöst. Das Skript muss überall dort verfügbar sein, wo Slurm-Befehle ausgeführt werden.

Kann ich das CLI-Filter-Plugin-Skript auf einem Live-Cluster ändern?

Ja, wenn Sie den S3- oder Dateisystembereitstellungsansatz verwenden. Neue Instances erhalten das aktualisierte Skript, aber für bestehende Instances muss das Skript manuell oder über die von Ihnen gewählte Bereitstellungsmethode aktualisiert werden.

Kann ich verschiedene CLI-Filter-Plugin-Skripte für verschiedene Compute-Knotengruppen verwenden?

Ja, aber das wird nicht empfohlen. Sie können Skripts mit unterschiedlicher Logik für verschiedene Rechenknotengruppen bereitstellen, aber Sie sind dafür verantwortlich, Interdependenzen zu verwalten und Logiküberschneidungen zu verhindern. Die meisten Kunden stellen einen Satz von Logik für einen gesamten Cluster bereit.

Kann ich das CLI-Filter-Plugin mit C-Implementierung anstelle von Lua verwenden?

Die C-Implementierung wird nicht unterstützt. Nur die Lua-Skriptimplementierung wird unterstützt in AWS STCK. SchedMD empfiehlt Kunden, Lua über C zu verwenden, um die Verwendung von CLI-Filter-Plugins zu vereinfachen.

Kann ich das CLI-Filter-Plugin in einem vorhandenen Cluster ein- oder ausschalten?

Ja, Sie können das CLI-Filter-Plugin auf vorhandenen Clustern mithilfe der Update API aktivieren oder deaktivieren, ohne den Cluster neu zu erstellen.

Behebung von Problemen mit dem Slurm CLI Filter Plugin in AWS 5 STCK

Verwenden Sie diese Informationen zur Fehlerbehebung, um häufig auftretende Probleme mit dem CLI-Filter-Plug-In zu lösen.

Die Auftragsübermittlung schlägt sofort fehl und es gibt einen Fehler beim Laden des Plugins

Symptome: Benutzer erhalten beim Senden von Aufträgen Fehlermeldungen über das fehlende oder fehlgeschlagene CLI-Filter-Plugin.

Mögliche Ursachen:

- Das CLI-Filter-Plugin-Skript fehlt auf einem oder mehreren Knoten
- Falscher Skriptdateiname (muss exakt sein `cli_filter.lua`)
- Das Skript wurde im falschen Verzeichnispfad bereitgestellt
- Das Skript hat falsche Dateiberechtigungen

Auflösung

- Stellen Sie sicher, dass das Skript unter existiert `/etc/aws/pcs/scheduler/slurm-`*`version`*`/cli_filter.lua` auf allen Anmelde- und Rechenknoten
- Überprüfen Sie, ob der Dateiname des Skripts genau ist `cli_filter.lua`
- Stellen Sie sicher, dass das Skript lesbare Berechtigungen hat (644 oder ähnlich)
- Testen Sie die Skriptbereitstellung auf einem einzelnen Anmeldeknoten, bevor Sie es im gesamten Cluster bereitstellen

Die Clustererstellung schlägt mit einem Validierungsfehler für das CLI Filter Plugin fehl

Symptome: Die Clustererstellung schlägt mit der Fehlermeldung „Ungültig“ fehl `CliFilterPlugins` Parameter.

Mögliche Ursachen:

- Falsches Parameterwertformat in `slurmCustomSettings`
- Tippfehler im Parameternamen oder -wert

Auflösung

- Verwenden Sie den exakten Parameternamen: `CliFilterPlugins`
- Verwenden Sie den exakten Parameterwert: `cli_filter/lua`
- Überprüfen Sie die JSON-Syntax in `slurmCustomSettings` Reihe

Das CLI-Filter-Plugin-Skript wird ausgeführt, aber die Jobvalidierung funktioniert nicht wie erwartet

Symptome: Jobs wurden erfolgreich übermittelt, aber die benutzerdefinierte Validierungslogik löst keine unerwarteten Ergebnisse aus oder erzeugt keine.

Mögliche Ursachen:

- Syntaxfehler im Lua-Skript
- Falsche Feldzugriffsmuster (unter Verwendung der Job Submit Plugin-Syntax anstelle des CLI-Filter-Plug-ins)
- Logikfehler bei den Validierungsbedingungen

Auflösung

- Überprüfen Sie das Lua-Skript auf Syntaxfehler
- Überprüfen Sie die Verwendungsmöglichkeiten des Feldzugriffs `options["field_name"]` formatieren statt `job_desc.field_name`
- Fügen Sie Protokollanweisungen zum Debug-Skriptausführungsablauf hinzu
- Testen Sie zuerst die Skriptlogik mit einfachen Validierungsfällen

Die Bereitstellung von S3-Skripten

Symptome: Instanzen werden gestartet, aber das CLI-Filter-Plugin-Skript wird nicht von S3 heruntergeladen.

Mögliche Ursachen:

- Dem IAM-Instanzprofil fehlen S3-Leseberechtigungen
- Der S3-VPC-Endpunkt ist nicht konfiguriert
- Falscher S3-Bucket oder Objektpfad in den Benutzerdaten

Auflösung

- Stellen Sie sicher, dass das IAM-Instanzprofil `s3:GetObject` Erlaubnis für deinen Bucket
- Konfigurieren Sie den S3 VPC Gateway-Endpunkt für den direkten Zugriff
- Überprüfen Sie den S3-Bucket-Namen und den Objektpfad im Benutzerdatenskript
- Überprüfen Sie die Benutzerdatenprotokolle der Instanz auf S3-Download-Fehler

Slurm Metriken in AWS STCK

AWS PCS unterstützt Slurm Metriken, die Clusterdaten in Echtzeit über HTTP-Endpunkte verfügbar machen, die kompatibel mit Prometheus und andere Überwachungssysteme. Einzelheiten, einschließlich Auswirkungen auf die Leistung und Sicherheitsaspekte, finden Sie in der Leitfaden für Kennzahlen auf der Slurm Webseite.

Voraussetzungen

Vor der Aktivierung Slurm Vergewissern Sie sich, dass Sie über folgende Kennzahlen verfügen:

- Cluster-Version: Slurm Version 25.11 oder höher.
- Sicherheitsgruppe: Regeln, die HTTP-Verkehr auf Port 6817 aus den von Ihnen gewünschten Quellen zulassen.

Aktivieren Sie den Metrik-Endpunkt

Legen Sie den folgenden benutzerdefinierten Wert auf Clusterebene fest Slurm Einstellungen:

- `MetricsType` — Muss ein unterstütztes Metrik-Plugin angeben, z. B. `metrics/openmetrics`.
- `CommunicationParameters` — Muss beinhalten `enable_http`.

> **Important**
>
> Aktivieren `enable_http` macht einen nicht authentifizierten HTTP-Endpunkt verfügbar. Jeder mit Netzwerkzugriff auf Port 6817 kann Cluster-, Job- und Node-Metriken lesen. Verwenden Sie Sicherheitsgruppenregeln, um den Zugriff nur auf vertrauenswürdige Quellen einzuschränken.

- `PrivateData` — Muss nicht gesetzt sein.

Für weitere Informationen zu Benutzerdefiniert Slurm Einstellungen finden Sie unter Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK.

Verwenden Sie den Metrik-Endpunkt

Fragen Sie den Metrik-Endpunkt von einem Host mit Netzwerkzugriff auf den Controller ab:

```
curl http://controller-ip:6817/metrics
```

Weitere Informationen zu verfügbaren Metriken und zur Scraping-Konfiguration finden Sie in der Leitfaden für Metriken auf der Slurm Webseite.

Um diese Metriken mithilfe eines verwalteten Systems zu sammeln Prometheus Collector, siehe Erfassen Slurm Metriken mit einem verwalteten Prometheus Kollektor.

Erfassen Slurm Metriken mit einem verwalteten Prometheus Kollektor

Nachdem Sie aktiviert haben Slurm Metriken auf Ihrem AWS PCS-Cluster (siehe Slurm Metriken in AWS STCK), können Sie einen verwalteten verwenden Prometheus Collector zum automatischen Scrappen der Metrik-Endpunkte und Bereitstellung der Daten für PromQL Abfragen. Der verwaltete Collector erstellt elastische Netzwerkschnittstellen in Ihren VPC-Subnetzen, um die Metrik-Endpunkte des Cluster-Controllers auf Port 6817 zu erreichen.

Der Collector kann Metriken an eines der folgenden Ziele liefern:

- Amazon Managed Service für Prometheus Workspace — Ein engagierter Prometheus-kompatibler Metrikspeicher mit konfigurierbarer Aufbewahrung (standardmäßig 150 Tage). Sie können Abfragen über Prometheus-kompatible APIs oder Grafana.
- CloudWatch Datensatz — Die Standardeinstellung Ihres Kontos CloudWatch Datensatz mit eingeschlossener Aufbewahrungsfrist von 15 Monaten. Sie können Abfragen über CloudWatch Query Studio oder das Prometheus-kompatible HTTP-API.

Die clusterseitige Konfiguration (Slurm Einstellungen, Controller-Ziel, Sicherheitsgruppen und Scrape-Konfiguration) sind identisch, unabhängig davon, welches Ziel Sie wählen. Nur der `destination` Block in der Create-Scraper-Anfrage und der Abfrageendpunkt unterscheiden sich.

Sie erstellen den Collector mit dem `aws amp create-scraper` Befehl. Obwohl dieser Befehl zu dem gehört `amp` Der CLI-Namespace unterstützt beide Ziele.

Für weitere Informationen über VPC-connected verwaltete Sammler finden Sie unter Richten Sie eine ein VPC-connected verwalteter Sammler in der Amazonas CloudWatch Benutzerleitfaden.

Voraussetzungen

Bevor Sie den verwalteten Collector konfigurieren, überprüfen Sie Folgendes:

- Slurm Metriken aktiviert — Der Metriken-Endpunkt muss in Ihrem Cluster aktiv sein. Sie aktivieren ihn, indem Sie den `MetricsType` und `CommunicationParameters` kundenspezifisch Slurm Einstellungen. Für Anweisungen zur Aktivierung Slurm Metriken finden Sie unter Slurm Metriken in AWS STCK. Für weitere Informationen über Benutzerdefiniert Slurm Einstellungen finden Sie unter Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK.

- Slurm Version 25.11 oder höher — Der Cluster muss laufen Slurm 25.11 oder höher, um den Metrik-Endpunkt verfügbar zu machen.
- Zielressource — Erstellen Sie das Ziel für Ihre Metriken:
  - Amazon Managed Service für Prometheus Workspace — Erstellen Sie einen Workspace und warten Sie, bis er ihn erreicht `ACTIVE` Status. Anweisungen zum Erstellen eines Arbeitsbereichs finden Sie unter Einen Arbeitsbereich erstellen in der Amazon Managed Service für Prometheus — Benutzerhandbuch.
  - CloudWatch Datensatz — Jedes Konto hat einen `default` Datensatz in jeder Region. Sie müssen es nicht erstellen.
- VPC-Subnetze und Netzwerke — Sie benötigen mindestens zwei Subnetze in verschiedenen Availability Zones innerhalb derselben VPC wie der Cluster-Controller. Schließen Sie die Availability Zone ein, in der sich die Netzwerkschnittstelle des Controllers befindet. Die VPC muss über DNS-Unterstützung und aktivierte DNS-Hostnamen verfügen. Detaillierte Netzwerkanforderungen finden Sie unter Richten Sie eine ein VPC-connected verwalteter Sammler in der Amazonas CloudWatch Benutzerleitfaden.
- Sicherheitsgruppen — Eine dedizierte Sicherheitsgruppe für den Collector ist erforderlich. Anweisungen zur Konfiguration von Sicherheitsgruppen finden Sie unter Konfigurieren Sie Sicherheitsgruppen für den Collector.
- IAM-Berechtigungen — Der IAM-Prinzipal, der die Scraper-Anforderungen erstellt `aps:CreateScraper` und `iam:CreateServiceLinkedRole` Genehmigungen. Der Dienst erstellt automatisch eine dienstverknüpfte Rolle (`AWSServiceRoleForAmazonPrometheusScraper`). Diese Rolle gewährt dem Collector die Erlaubnis, auf Ihre VPC-Ressourcen zuzugreifen und an das von Ihnen gewählte Ziel zu schreiben. Es ist keine manuelle Rolleneinrichtung erforderlich. Weitere Informationen finden Sie unter Verwenden von dienstverknüpften Rollen in der Amazon Managed Service für Prometheus — Benutzerhandbuch.
- Internet- oder VPC-Endpunkt — Die Collector-Subnetze müssen in der Lage sein, den Zieldienst zu erreichen. Wenn Ihre Subnetze keinen Internetzugang haben, erstellen Sie einen Schnittstellen-VPC-Endpunkt in derselben VPC und denselben Subnetzen. Verwenden `com.amazonaws.`*`region`*`.aps-workspaces` für eine Workspace-Destination von Amazon Managed Service für Prometheus, oder `com.amazonaws.`*`region`*`.monitoring` für einen CloudWatch Ziel des Datensatzes.

Konfigurieren Sie Sicherheitsgruppen für den Collector

Wir empfehlen, eine dedizierte Sicherheitsgruppe für den verwalteten Collector zu erstellen, anstatt eine vorhandene Sicherheitsgruppe wiederzuverwenden. Eine dedizierte Gruppe gibt Ihnen explizite, überprüfbare Regeln, die dem Prinzip der geringsten Rechte folgen.

> **Important**
>
> Die `enable_http` Die Einstellung macht einen nicht authentifizierten HTTP-Endpunkt auf Port 6817 verfügbar. Beschränken Sie den eingehenden Zugriff auf diesen Port nur auf die Sicherheitsgruppe des Collectors. Erlauben Sie keinen breiten Netzwerkzugriff (z. B. einen CIDR-Bereich) auf diesen Port.

Das folgende Verfahren verwendet Shell-Variablen für Sicherheitsgruppen-IDs. Legen Sie diese Variablen fest, bevor Sie die Befehle ausführen:

- `VPC_ID` — Die ID der VPC, in der Ihr AWS Der PCS-Cluster befindet sich.
- `CLUSTER_SG_ID` — Die ID der Sicherheitsgruppe, die an Ihren Cluster angehängt ist (die, die Sie bei der Erstellung des Clusters angegeben haben).
- `VPCE_SG_ID` — Die ID der Sicherheitsgruppe, die an Ihren Schnittstellen-VPC-Endpunkt angehängt ist. Diese Variable wird nur benötigt, wenn Ihre Collector-Subnetze den Zieldienst über einen Schnittstellen-VPC-Endpunkt und nicht über Internet oder NAT-Ausgang erreichen. Der Endpunkt ist `com.amazonaws.`*`region`*`.aps-workspaces` für ein Amazon Managed Service for Prometheus Workspace-Ziel oder `com.amazonaws.`*`region`*`.monitoring` für ein CloudWatch Ziel des Datensatzes.

Um Sicherheitsgruppen für die verwalteten Benutzer zu konfigurieren Prometheus Kollektor

1. Erstellen Sie eine dedizierte Sicherheitsgruppe für den Collector und erfassen Sie die Gruppen-ID:

```
COLLECTOR_SG_ID=$(aws ec2 create-security-group \
    --group-name "pcs-prometheus-collector" \
    --description "Security group for managed Prometheus collector" \
    --vpc-id "$VPC_ID" \
    --query 'GroupId' \
    --output text)
```

2. Fügen Sie der Sicherheitsgruppe des Clusters eine Regel für eingehenden Datenverkehr hinzu, die TCP-Verkehr auf Port 6817 aus der Sicherheitsgruppe des Collectors zulässt. Diese Regel ermöglicht es dem Collector, das zu löschen Slurm Endpunkt der Metriken auf dem Controller.

   Benutze den `--ip-permissions` Formular, um eine Regelbeschreibung aus Gründen der Überprüfbarkeit einzufügen:

   ```
   aws ec2 authorize-security-group-ingress \
       --group-id "$CLUSTER_SG_ID" \
       --ip-permissions \

    'IpProtocol=tcp,FromPort=6817,ToPort=6817,UserIdGroupPairs=[{GroupId='"$COLLECTOR_SG_ID"',
    collector scrapes Slurm metrics"}]'
   ```

   Alternativ können Sie das kürzere Formular ohne eine Regelbeschreibung verwenden:

   ```
   aws ec2 authorize-security-group-ingress \
       --group-id "$CLUSTER_SG_ID" \
       --protocol tcp \
       --port 6817 \
       --source-group "$COLLECTOR_SG_ID"
   ```

3. (Optional) Sperren Sie den ausgehenden Datenverkehr in der Sicherheitsgruppe des Collectors. Standardmäßig lässt eine neu erstellte Sicherheitsgruppe den gesamten ausgehenden Verkehr zu. Wenn Sie aus Gründen einer höheren Sicherheitslage nur expliziten ausgehenden Datenverkehr erzwingen möchten, heben Sie die Standardregel für alle Zugriffe auf und fügen Sie nur die Regeln für ausgehenden Datenverkehr hinzu, die der Collector benötigt.

   Widerrufen Sie die Standardregel, die allen ausgehenden Verkehr zulässt:

   ```
   aws ec2 revoke-security-group-egress \
       --group-id "$COLLECTOR_SG_ID" \
       --ip-permissions '[{"IpProtocol":"-1","IpRanges":[{"CidrIp":"0.0.0.0/0"}]}]'
   ```

   Fügen Sie eine explizite Regel für ausgehenden Datenverkehr hinzu, damit der Collector den Controller über TCP 6817 erreichen kann:

   ```
   aws ec2 authorize-security-group-egress \
       --group-id "$COLLECTOR_SG_ID" \
       --ip-permissions \
   ```

```
 'IpProtocol=tcp,FromPort=6817,ToPort=6817,UserIdGroupPairs=[{GroupId='"$CLUSTER_SG_ID"',De
 to Slurm controller for metrics scraping"}]'
```

Fügen Sie eine explizite Ausgangsregel für HTTPS (TCP 443) hinzu, um das Ziel der Metrikzustellung zu erreichen (Amazon Managed Service für Prometheus oder CloudWatch):

```
aws ec2 authorize-security-group-egress \
    --group-id "$COLLECTOR_SG_ID" \
    --ip-permissions \

 'IpProtocol=tcp,FromPort=443,ToPort=443,IpRanges=[{CidrIp=0.0.0.0/0,Description="HTTPS
 egress for metrics delivery"}]'
```

Für eine strengere Kontrolle in Subnetzen, die einen VPC-Endpunkt für die Übertragung verwenden, ersetzen Sie den CIDR-Bereich durch die Sicherheitsgruppe des VPC-Endpunkts:

```
aws ec2 authorize-security-group-egress \
    --group-id "$COLLECTOR_SG_ID" \
    --ip-permissions \

 'IpProtocol=tcp,FromPort=443,ToPort=443,UserIdGroupPairs=[{GroupId='"$VPCE_SG_ID"',Descrip
 egress to VPC endpoint for metrics delivery"}]'
```

> **Note**
>
> Wenn Sie die Standardregel für ausgehenden Datenverkehr nicht aufgehoben haben, können Sie diesen Schritt überspringen. Die Standardregel lässt bereits den gesamten ausgehenden Verkehr zu, einschließlich des Datenverkehrs zu Port 6817 und Port 443.

4. (Isolierte Cluster) Wenn Ihre Collector-Subnetze keinen Internetzugang haben und einen Schnittstellen-VPC-Endpunkt verwenden, um den Zieldienst zu erreichen, fügen Sie der Sicherheitsgruppe des VPC-Endpunkts eine Regel für eingehenden Datenverkehr hinzu. Diese Regel ermöglicht es dem HTTPS-Verkehr des Collectors, die Netzwerkschnittstellen des Endpunkts zu erreichen. Dies ist ein häufig übersehener Schritt.

```
aws ec2 authorize-security-group-ingress \
    --group-id "$VPCE_SG_ID" \
    --ip-permissions \
```

```
'IpProtocol=tcp,FromPort=443,ToPort=443,UserIdGroupPairs=[{GroupId='"$COLLECTOR_SG_ID"',De
collector reaches service endpoint"}]'
```

> **Note**
>
> Als einfachere, aber weniger restriktive Alternative können Sie die vorhandene Sicherheitsgruppe des Controllers an den Collector anhängen, wenn diese Sicherheitsgruppe eine selbstreferenzierende Regel enthält, die den Datenverkehr von sich selbst zulässt. Dadurch wird die Konnektivitätsanforderung erfüllt, ohne dass eine dedizierte Gruppe erstellt werden muss.
> Zum Beispiel, wenn die Sicherheitsgruppe Ihres Clusters (`sg-0abc1234def56789a`) lässt bereits den gesamten TCP-Verkehr von sich aus zu, übergeben Sie diese Sicherheitsgruppen-ID im `--security-group-ids` Parameter beim Erstellen des Scrapers:
>
> ```
> aws amp create-scraper \
>     --source '{"vpcConfiguration":{"subnetIds":["subnet-id-1","subnet-
> id-2"],"securityGroupIds":["sg-0abc1234def56789a"]}}' \
>     ...
> ```
>
> Der im vorherigen Verfahren beschriebene Ansatz für dedizierte Sicherheitsgruppen bietet jedoch eine höhere Sicherheitsbarriere mit expliziten, überprüfbaren Regeln.

Weitere Informationen zu den Regeln der Sicherheitsgruppe finden Sie unter Sicherheitsgruppenregeln im Amazon VPC-Benutzerhandbuch.

Erstellen Sie den verwalteten Collector

Verwenden Sie den AWS CLI um eine zu erstellen VPC-connected verwalteter Collector, der Ihren Cluster-Controller scrapft und Metriken an das von Ihnen gewählte Ziel liefert.

Um einen verwalteten Collector zu erstellen für Slurm Metriken

1. Speichern Sie die Scrape-Konfiguration in einer lokalen YAML-Datei mit dem Namen `scrape-config.yaml`. Eine empfohlene Konfiguration finden Sie unter Vorgeschlagene Scrape-Konfiguration.

2. Erstellen Sie eine JSON-Eingabedatei mit dem Namen `create-scraper-input.json` mit der folgenden Struktur. Wählen Sie den `destination` Block, der zu Ihrem Ziel passt.

```
{
  "source": {
    "vpcConfiguration": {
      "subnetIds": ["subnet-1", "subnet-2"],
      "securityGroupIds": ["sg-collector"]
    }
  },
  "destination": { ... },
  "scrapeConfiguration": {
    "configurationBlob": "raw-yaml-contents"
  }
}
```

Ersetzen Sie:

- *subnet-1* und *subnet-2* — Mindestens zwei Subnetz-IDs in verschiedenen Availability Zones innerhalb derselben VPC wie der Cluster-Controller.
- *sg-collector* — Die Sicherheitsgruppen-ID für den verwalteten Collector.
- *raw-yaml-contents* — Der vollständige Text Ihres `scrape-config.yaml` Datei, eingefügt als einzelner JSON-Zeichenfolgenwert. Die AWS CLI Base64 kodiert den Wert auf der Leitung automatisch.

Für den `destination` Verwenden Sie eines der folgenden Felder:

Ziel: Amazon Managed Service für Prometheus Workspace

```
"destination": {
  "ampConfiguration": {
    "workspaceArn": "arn:aws:aps:region:account-id:workspace/workspace-id"
  }
}
```

Zielort: CloudWatch Datensatz

```
"destination": {
  "cloudWatchConfiguration": {
```

```
    "datasetArn": "arn:aws:cloudwatch:region:account-id:dataset/default"
  }
}
```

Den vollständigen Parametersatz finden Sie unter create-scraper in der AWS CLI Befehlsreferenz.

3. Erstellen Sie den Scraper:

```
aws amp create-scraper --cli-input-json file://create-scraper-input.json
```

Der Befehl gibt eine zurück `scraperId` und einen Status von `CREATING`.

4. Warten Sie, bis sich der Scraper-Status zu ändert `ACTIVE` (normalerweise 5—15 Minuten):

```
aws amp describe-scraper --scraper-id scraper-id
```

Ersetzen *`scraper-id`* mit der im vorherigen Schritt zurückgegebenen ID. Sie können einen Scraper erst löschen, wenn er erreicht ist `ACTIVE` Status.

Finden Sie den Cluster-Controller-Endpunkt

Die Scrape-Konfiguration erfordert die private IP-Adresse Ihres AWS PCS-Cluster-Controller. Verwenden Sie eine der folgenden Methoden, um ihn zu finden.

AWS-Managementkonsole

1. Öffnen Sie das AWS PCS-Konsole unter https://console.aws.amazon.com/pcs/.
2. Wählen Sie Ihren Cluster aus der Liste aus.
3. Suchen Sie in den Cluster-Konfigurationsdetails nach Endpunkte Abschnitt.
4. Notieren Sie sich die private IP-Adresse und den Port für Slurm Controller (slurmctld). Der Port ist 6817.

AWS CLI

1. Führen Sie den folgenden Befehl aus. Ersetzen *`cluster-identifier`* mit Ihrem Clusternamen oder Ihrer Cluster-ID.

```
aws pcs get-cluster --cluster-identifier cluster-identifier
```

Suchen Sie in der Antwort nach `SLURMCTLD` Eintrag in der `endpoints` Reihe. Das `privateIpAddress` Der Wert ist der Controller-Endpunkt, den Sie für die Scrape-Konfiguration benötigen. Ein Beispiel:

```
"endpoints": [
    {
        "type": "SLURMCTLD",
        "privateIpAddress": "192.0.2.1",
        "port": "6817"
    },
    {
        "type": "SLURMRESTD",
        "privateIpAddress": "192.0.2.1",
        "port": "6820"
    }
]
```

Benutze den `privateIpAddress` von der `SLURMCTLD` Eintrag (Port 6817) als *`controller-endpoint`* Wert in Ihrer Scrape-Konfiguration.

2. Extrahieren Sie alternativ nur die Controller-IP-Adresse direkt:

```
aws pcs get-cluster --cluster-identifier cluster-identifier \
    --query 'cluster.endpoints[?type==`SLURMCTLD`].privateIpAddress' \
    --output text
```

Vorgeschlagene Scrape-Konfiguration

Die folgende YAML-Konfiguration scrappt vier Slurm Metrik-Endpunkte (Jobs, Knoten, Scheduler und Partitionen) vom Cluster-Controller aus. Jeder Endpunkt ist als separater Job definiert, sodass Sie Metriken in Ihren Abfragen anhand der Quelle identifizieren können.

```
global:
  scrape_interval: 60s
  scrape_timeout: 30s

scrape_configs:
```

```
  - job_name: 'slurm-jobs'
    metrics_path: /metrics/jobs
    static_configs:
      - targets:
          - 'controller-endpoint:6817'
    relabel_configs:
      - target_label: cluster
        replacement: 'my-cluster-name'

  - job_name: 'slurm-nodes'
    metrics_path: /metrics/nodes
    static_configs:
      - targets:
          - 'controller-endpoint:6817'
    relabel_configs:
      - target_label: cluster
        replacement: 'my-cluster-name'

  - job_name: 'slurm-scheduler'
    metrics_path: /metrics/scheduler
    static_configs:
      - targets:
          - 'controller-endpoint:6817'
    relabel_configs:
      - target_label: cluster
        replacement: 'my-cluster-name'

  - job_name: 'slurm-partitions'
    metrics_path: /metrics/partitions
    static_configs:
      - targets:
          - 'controller-endpoint:6817'
    relabel_configs:
      - target_label: cluster
        replacement: 'my-cluster-name'
```

Ersetzen Sie:

- *`controller-endpoint`* — Die private IP-Adresse Ihres AWS PCS-Cluster-Controller. Anweisungen zum Ermitteln dieses Werts finden Sie unter Finden Sie den Cluster-Controller-Endpunkt.

- *my-cluster-name* — Eine Bezeichnung, die Ihren Cluster identifiziert. Das `relabel_configs` Blockstempel a `cluster` Etikett auf jeder Metrik dieses Scrapers. Filtern Sie Abfragen immer auf dem `cluster` Etikett. Serien von einem Scraper, bei denen das Etikett nicht gestempelt wurde, werden als doppelte unbeschriftete Serien angezeigt, bis sie veraltet sind.

Das Minimum `scrape_interval` für einen verwalteten Kollektor sind es 30 Sekunden. Bei dieser Konfiguration werden 60 Sekunden benötigt, da beim Scraping die Slurm Controller (slurmctld). Beim Abfragen von Metriken werden interne Sperren erfasst und speicherinterne Datenstrukturen gelesen. Diese Aktivität kann die Leistung des Schedulers auf stark frequentierten Clustern beeinträchtigen. Der Slurm Der Metrics Guide empfiehlt ein Scrape-Intervall von 60—120 Sekunden, um die Auswirkungen auf die Leistung so gering wie möglich zu halten.

> **Note**
>
> Diese Konfiguration beinhaltet nicht `/metrics/jobs-users-accts` Endpunkt. Slurm Die Dokumentation warnt davor, dass dieser Endpunkt eine unbegrenzte Anzahl von Serien erzeugt und für die gespeicherte Überwachung ungeeignet ist. Kratzen Sie nicht den Rohling ab `/metrics` auch der Index, da alle Unterendpunktdaten in einer einzigen Antwort zusammengefasst werden.

Weitere Informationen zu den unterstützten Scrape-Konfigurationsoptionen finden Sie unter Scraper-Konfiguration in der Amazonas CloudWatch Benutzerleitfaden.

Überprüfen Sie die Bereitstellung der Kennzahlen

Nachdem der Schaber erreicht ist `ACTIVE` Status, die ersten Datenpunkte erscheinen ungefähr ein Scrape-Intervall zuzüglich einer späteren Lieferzeit. Verwenden Sie die Bestätigungsmethode, die Ihrem Ziel entspricht.

Bestätigen Sie die Lieferung an einen Amazon Managed Service für Prometheus Workspace

Sende eine SigV4-signed Anfrage, die verfügbaren Metriknamen aus deinem Workspace aufzulisten:

```
awscurl --service aps --region region \
    "https://aps-workspaces.region.amazonaws.com/workspaces/workspace-id/api/v1/label/
__name__/values"
```

Der anrufende Prinzipal benötigt `aps:QueryMetrics` und `aps:GetLabels` Berechtigungen (oder die `AmazonPrometheusQueryAccess` verwaltete Richtlinie).

Überprüfen Sie die Lieferung an einen CloudWatch Datensatz

Senden Sie einen SigV4-signed Anfrage zur Auflistung der verfügbaren Metriknamen von Ihrem CloudWatch Datensatz:

```
awscurl --service monitoring --region region \
    "https://monitoring.region.amazonaws.com/api/v1/label/__name__/values"
```

Der anrufende Principal benötigt `cloudwatch:GetMetricData` und `cloudwatch:ListMetrics` Genehmigungen.

> **Important**
>
> Metriken, die an einen geliefert werden CloudWatch Datensätze werden gespeichert als OpenTelemetry (Otel) -Metriken. Das tun sie nicht erscheinen im Klassiker CloudWatch Metrik-Namespace-Browser oder in der Ausgabe von `aws cloudwatch list-metrics`. Sie müssen sie abfragen mit PromQL.

Suchen Sie für beide Ziele nach Metriknamen, die mit beginnen `slurm_`, wie `slurm_nodes`, `slurm_jobs_running`, oder `slurm_node_cpus`. Falls nein Slurm Wenn Metriken angezeigt werden, überprüfen Sie Folgendes:

- Der Scraper-Status ist `ACTIVE`.
- Sicherheitsgruppenregeln ermöglichen es dem Collector, Port 6817 auf dem Controller zu erreichen.
- Der Slurm Der Metrik-Endpunkt ist auf dem Cluster aktiviert.

Query Slurm Metriken mit PromQL

Sie fragen das gesammelte ab Slurm Metriken verwenden PromQL. Dieselben Abfragen funktionieren für beide Ziele. Die Abfragemethode hängt davon ab, wo Sie die Metriken bereitgestellt haben.

Fragen Sie einen Amazon Managed Service für Prometheus Workspace ab

- HTTP-API — Senden SigV4-signed Anfragen (Dienstname) `aps`) zu `https://aps-workspaces.region.amazonaws.com/workspaces/workspace-id/api/v1/query` oder `/api/v1/query_range`.
- Grafana — Füge eine hinzu Prometheus Datenquelle mit SigV4-Authentifizierung und Dienstname `aps`. Für Anweisungen zum Abfragen mit Grafana, siehe Abfrage mit Grafana in der Amazon Managed Service für Prometheus — Benutzerhandbuch.

Das folgende Beispiel verwendet `awscurl` um laufende Jobs von einem Amazon Managed Service for Prometheus Workspace abzufragen:

```
awscurl --service aps --region region \
    -X POST "https://aps-workspaces.region.amazonaws.com/workspaces/workspace-id/api/
v1/query" \
    -H "Content-Type: application/x-www-form-urlencoded" \
    -d "query=slurm_jobs_running"
```

Der aufrufende Prinzipal benötigt `aps:QueryMetrics`, `aps:GetMetricMetadata`, `aps:GetSeries`, und `aps:GetLabels` Berechtigungen (oder `AmazonPrometheusQueryAccess` verwaltete Richtlinie).

Abfrage a CloudWatch Datensatz

- CloudWatch Konsole — Öffnen CloudWatch, wähle Query Studio, und wählen Sie PromQL aus dem Menü der Abfragesprache.
- HTTP API — Senden SigV4-signed Anfragen (Dienstname) `monitoring`) zu `https://monitoring.region.amazonaws.com/api/v1/query` oder `/api/v1/query_range`.
- Grafana — Füge eine hinzu Prometheus Datenquelle mit URL `https://monitoring.region.amazonaws.com`, SigV4-Authentifizierung und Dienstname `monitoring`. Für Anweisungen zum Abfragen CloudWatch Metriken mit PromQL in Grafana, siehe Abfrage CloudWatch Metriken mit PromQL in Grafana in der Amazon Managed Grafana-Benutzerhandbuch.

Das folgende Beispiel verwendet `awscurl` um laufende Jobs von einem abzufragen CloudWatch Datensatz:

```
awscurl --service monitoring --region region \
    -X POST "https://monitoring.region.amazonaws.com/api/v1/query" \
    -H "Content-Type: application/x-www-form-urlencoded" \
    -d "query=slurm_jobs_running"
```

Der anrufende Principal benötigt `cloudwatch:GetMetricData` und `cloudwatch:ListMetrics` Genehmigungen.

Beispiel PromQL queries

Die folgenden Abfragen funktionieren für beide Ziele. Ersetzen *`my-cluster-name`* mit dem Wert, den Sie in Ihrer Scrape-Konfiguration festgelegt haben `cluster` umetikettieren.

Prozentsatz der CPU-Auslastung

```
100 * slurm_node_cpus_alloc{cluster="my-cluster-name"} /
 slurm_node_cpus{cluster="my-cluster-name"}
```

Ausstehende Jobs (Warteschlangenrückstand)

```
slurm_jobs_pending{cluster="my-cluster-name"}
```

Ausführen von Aufgaben

```
slurm_jobs_running{cluster="my-cluster-name"}
```

Auftragsdurchsatz

```
slurm_jobs_completed{cluster="my-cluster-name"}
```

Weitere Informationen zu verfügbaren Metriken und zur Scraping-Konfiguration finden Sie im Leitfaden für Metriken auf der Slurm Webseite.

Sicherheit bei AWS Dienst für parallele Datenverarbeitung

Cloud-Sicherheit bei AWS hat höchste Priorität. Als AWS Als Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die so konzipiert sind, dass sie die Anforderungen der sicherheitsrelevantesten Organisationen erfüllen.

Sicherheit ist eine gemeinsame Verantwortung zwischen AWS und du. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der laufenden Infrastruktur AWS Dienste in der AWS Cloud. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-party Auditoren testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheit im Rahmen der AWS Compliance-Programme. Um mehr über die Compliance-Programme zu erfahren, die gelten für AWS Parallel Computing Service finden Sie unter AWS Dienstleistungen im Rahmen des Compliance-Programms.
- Sicherheit in der Cloud — Ihre Verantwortung wird bestimmt durch AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung anwenden können, wenn Sie AWS 5 STCK. Die folgenden Themen zeigen Ihnen, wie Sie konfigurieren AWS PCS zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele. Sie lernen auch, andere zu verwenden AWS Dienste, die Ihnen helfen, Ihre zu überwachen und zu sichern AWS PCS-Ressourcen.

Topics

- Datenschutz in AWS Dienst für parallele Datenverarbeitung
- Zugriff AWS Parallel Computing Service mithilfe eines Schnittstellenendpunkts (AWS PrivateLink)
- Identitäts- und Zugriffsmanagement für AWS Dienst für parallele Datenverarbeitung
- Überprüfung der Konformität für AWS Dienst für parallele Datenverarbeitung
- Widerstandsfähigkeit in AWS Dienst für parallele Datenverarbeitung
- Sicherheit der Infrastruktur in AWS Dienst für parallele Datenverarbeitung
- Analyse und Management von Sicherheitslücken in AWS Dienst für parallele Datenverarbeitung
- Cross-service verwirrte stellvertretende Prävention

- Bewährte Sicherheitsmethoden für AWS Dienst für parallele Datenverarbeitung

Datenschutz in AWS Dienst für parallele Datenverarbeitung

Das AWS Modell der gemeinsamen Verantwortung gilt für den Datenschutz in AWS Dienst für parallele Datenverarbeitung. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum DatenschutzRechtliche. Weitere Informationen zum Datenschutz in Europa finden Sie im Zentrum für die Datenschutz-Grundverordnung (DSGVO).

Aus Datenschutzgründen empfehlen wir Ihnen, Folgendes zu schützen AWS-Konto Anmeldeinformationen und richten Sie einzelne Benutzer ein mit AWS IAM Identity Center oder AWS Identity and Access Management (ICH BIN). So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Benutzen SSL/TLS um zu kommunizieren mit AWS Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten ein mit AWS CloudTrail. Für Informationen zur Verwendung CloudTrail Spuren zum Erfassen AWS Aktivitäten, siehe Arbeiten mit CloudTrail Wanderwege in der AWS CloudTrail Benutzerleitfaden.
- Benutzen AWS Verschlüsselungslösungen sowie alle darin enthaltenen Standard-Sicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie beim Zugriff auf FIPS 140-3-validierte kryptografische Module benötigen AWS Verwenden Sie über eine Befehlszeilenschnittstelle oder eine API einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit

arbeiten AWS PCS oder anderes AWS-Services mit der Konsole, API, AWS CLI, oder AWS SDKs. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

Die Verschlüsselung ist standardmäßig für ruhende Daten aktiviert, wenn Sie eine erstellen AWS Dienst für parallele Datenverarbeitung (AWS PCS) -Cluster mit dem AWS-Managementkonsole, AWS CLI, AWS PCS API, oder AWS SDKs. AWS PCS verwendet eine AWS eigener KMS-Schlüssel um Daten im Ruhezustand zu verschlüsseln. Weitere Informationen finden Sie unter Kundenschlüssel und AWS Schlüssel in der AWS KMS Leitfaden für Entwickler. Sie können auch einen vom Kunden verwalteten Schlüssel verwenden. Weitere Informationen finden Sie unter Erforderliche KMS-Schlüsselrichtlinie für die Verwendung mit verschlüsselten EBS-Volumes in AWS STCK.

Der Geheimer Cluster-Schlüssel ist gespeichert in AWS Secrets Manager und ist mit dem von Secrets Manager verwalteten KMS-Schlüssel verschlüsselt. Weitere Informationen finden Sie unter Arbeiten mit Cluster-Secrets in AWS STCK.

In einem AWS PCS-Cluster, die folgenden Daten sind im Ruhezustand:

- Status des Schedulers — Es enthält Daten zu laufenden Jobs und bereitgestellten Knoten im Cluster. Das sind die Daten, die Slurm persistiert `StateSaveLocation` definiert in Ihrem `slurm.conf`. Weitere Informationen finden Sie in der Beschreibung von StateSaveLocation in der Slurm-Dokumentation. AWS PCS löscht Jobdaten, nachdem ein Job abgeschlossen ist.
- Geheim für die Scheduler-Authentifizierung — AWS PCS verwendet es, um die gesamte Scheduler-Kommunikation im Cluster zu authentifizieren.

Informationen zum Scheduler-Status finden Sie unter AWS PCS verschlüsselt Daten und Metadaten automatisch, bevor es sie in das Dateisystem schreibt. Das verschlüsselte Dateisystem verwendet den Industriestandard AES-256 Verschlüsselungsalgorithmus für ruhende Daten.

Verschlüsselung während der Übertragung

Ihre Verbindungen zum AWS Die PCS-API verwendet die TLS-Verschlüsselung mit dem Signature Version 4-Signaturverfahren, unabhängig davon, ob Sie das verwenden AWS Command Line Interface (AWS CLI) oder AWS SDKs. Weitere Informationen finden Sie unter Signieren AWS

API-Anfragen in der AWS Identity and Access Management Benutzerleitfaden. AWS verwaltet die Zugriffskontrolle über die API mit den IAM-Richtlinien für die Sicherheitsanmeldeinformationen, die Sie für die Verbindung verwenden.

AWS PCS verwendet TLS, um eine Verbindung zu anderen herzustellen AWS Dienste.

Innerhalb eines Slurm-Clusters ist der Scheduler mit dem konfiguriert `auth/slurm` Authentifizierungs-Plug-In, das die Authentifizierung für die gesamte Scheduler-Kommunikation ermöglicht. Slurm bietet keine Verschlüsselung auf Anwendungsebene für seine Kommunikation. Alle Daten, die über Cluster-Instances übertragen werden, bleiben lokal in der EC2-VPC und unterliegen daher der VPC-Verschlüsselung, wenn diese Instances Verschlüsselung bei der Übertragung unterstützen. Weitere Informationen finden Sie unter Verschlüsselung bei der Übertragung in der Amazon Elastic Compute Cloud-Benutzerhandbuch. Die Kommunikation zwischen dem Controller (bereitgestellt in einem Dienstkonto) und den Clusterknoten in Ihrem Konto ist verschlüsselt.

Schlüsselverwaltung

AWS PCS verwendet eine AWS eigener KMS-Schlüssel um Daten zu verschlüsseln. Weitere Informationen finden Sie unter Kundenschlüssel und AWS Schlüssel in der AWS KMS Leitfaden für Entwickler. Sie können auch einen vom Kunden verwalteten Schlüssel verwenden. Weitere Informationen finden Sie unter Erforderliche KMS-Schlüsselrichtlinie für die Verwendung mit verschlüsselten EBS-Volumes in AWS STCK.

Der Geheimer Cluster-Schlüssel ist gespeichert in AWS Secrets Manager und ist mit dem von Secrets Manager verwalteten KMS-Schlüssel verschlüsselt. Weitere Informationen finden Sie unter Arbeiten mit Cluster-Secrets in AWS STCK.

Inter-network Datenschutz im Verkehr

AWS Die PCS-Rechenressourcen für einen Cluster befinden sich in einer VPC im Kundenkonto. Daher sind alle internen AWS Der PCS-Dienstverkehr innerhalb eines Clusters bleibt innerhalb von AWS Netzwerk und wird nicht über das Internet übertragen. Kommunikation zwischen dem Benutzer und AWS PCS-Knoten können über das Internet übertragen werden. Wir empfehlen, SSH oder Systems Manager zu verwenden, um eine Verbindung zu den Knoten herzustellen. Weitere Informationen finden Sie unter Was ist AWS Systems Manager? in der AWS Systems Manager Benutzerleitfaden.

Sie können auch die folgenden Angebote verwenden, um Ihr lokales Netzwerk mit zu verbinden AWS:

- AWS Site-to-Site VPN. Weitere Informationen finden Sie unter Was ist AWS Site-to-Site VPN? in der AWS Site-to-Site VPN Benutzerleitfaden.
- Ein AWS Direct Connect. Weitere Informationen finden Sie unter Was ist AWS Direct Connect? in der AWS Direct Connect Benutzerleitfaden.

Sie greifen auf das zu AWS PCS-API zur Ausführung administrativer Aufgaben für den Dienst. Sie und Ihre Benutzer greifen auf die Slurm-Endpunkt-Ports zu, um direkt mit dem Scheduler zu interagieren.

Verschlüsselung des API-Datenverkehrs

Um auf das zuzugreifen AWS Die PCS-API muss von den Clients Transport Layer Security (TLS) 1.2 oder höher unterstützt werden. Wir benötigen TLS 1.2 und empfehlen TLS 1.3. Kunden müssen außerdem Verschlüsselungssammlungen mit Perfect Forward Secrecy (PFS) unterstützen, wie z. B. Ephemeral Diffie-Hellman (DHE) oder Elliptische Kurve Diffie-Hellman Kurzlebig (ECDHE). Die meisten modernen Systemen 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. Sie können auch verwenden AWS -Security-Token-Service (AWS STS), um temporäre Sicherheitsanmeldeinformationen zum Signieren von Anfragen zu generieren.

Verschlüsselung des Datenverkehrs

Die Verschlüsselung von Daten während der Übertragung wird von unterstützten EC2-Instances aus aktiviert, die auf den Scheduler-Endpunkt zugreifen, und zwischen ComputeNodeGroup Instanzen aus dem AWS Cloud. Weitere Informationen finden Sie unter Verschlüsselung während der Übertragung.

Erforderliche KMS-Schlüsselrichtlinie für die Verwendung mit verschlüsselten EBS-Volumes in AWS STCK

AWS PCS verwendet dienstlich verknüpfte Rollen um Berechtigungen an andere zu delegieren AWS-Services. Das AWS Die dienstverknüpfte Rolle von PCS ist vordefiniert und umfasst folgende Berechtigungen AWS PCS muss andere anrufen AWS-Services in Ihrem Namen. Die vordefinierten Berechtigungen beinhalten auch den Zugriff auf Ihre Von AWS verwaltete Schlüssel aber nicht auf Ihre vom Kunden verwalteten Schlüssel.

In diesem Thema wird beschrieben, wie Sie die Schlüsselrichtlinie einrichten, die zum Starten von Instances erforderlich ist, wenn Sie einen vom Kunden verwalteten Schlüssel für die Amazon EBS-Verschlüsselung angeben.

> **Note**
>
> AWS PCS benötigt keine zusätzliche Autorisierung, um die Standardeinstellung zu verwenden Von AWS verwalteter Schlüssel um die verschlüsselten Volumes in Ihrem Konto zu schützen.

Inhalt

- -Übersicht
- Konfigurieren von Schlüsselrichtlinien
- Beispiel 1: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel erlauben
- Beispiel 2: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel über mehrere Konten erlauben
- Bearbeiten Sie die wichtigsten Richtlinien in der AWS KMS Konsole

-Übersicht

Sie können Folgendes verwenden AWS KMS keys für die Amazon EBS-Verschlüsselung wenn AWS PCS startet Instances:

- Von AWS verwalteter Schlüssel — Ein Verschlüsselungsschlüssel in Ihrem Konto, den Amazon EBS erstellt, besitzt und verwaltet. Dies ist der Standardverschlüsselungsschlüssel für ein neues Konto. Amazon EBS verwendet den Von AWS verwalteter Schlüssel für die Verschlüsselung, es sei denn, Sie geben einen vom Kunden verwalteten Schlüssel an.
- Kundenverwalteter Schlüssel – Ein benutzerdefinierter Verschlüsselungsschlüssel, den Sie erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter Erstellen Sie einen KMS-Schlüssel in der AWS Key Management Service Leitfaden für Entwickler.

> **Note**
>
> Der Schlüssel muss symmetrisch sein. Amazon EBS unterstützt keine asymmetrischen, vom Kunden verwalteten Schlüssel.

Sie konfigurieren vom Kunden verwaltete Schlüssel, wenn Sie verschlüsselte Snapshots oder eine Startvorlage erstellen, die verschlüsselte Volumes spezifiziert, oder wenn Sie die Verschlüsselung standardmäßig aktivieren.

Konfigurieren von Schlüsselrichtlinien

Ihre KMS-Schlüssel müssen über eine Schlüsselrichtlinie verfügen, die Folgendes ermöglicht AWS PCS zum Starten von Instances mit Amazon EBS-Volumes, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt wurden.

Verwenden Sie die Beispiele auf dieser Seite, um eine wichtige Richtlinie zu konfigurieren, die Folgendes bietet AWS PCS-Zugriff auf Ihren vom Kunden verwalteten Schlüssel. Sie können die Schlüsselrichtlinie für den vom Kunden verwalteten Schlüssel bei der Erstellung des Schlüssels oder zu einem späteren Zeitpunkt ändern.

Die Schlüsselrichtlinie muss die folgenden Aussagen enthalten:

- Eine Anweisung, die die IAM-Identität zulässt, die in der `Principal` Element, um den vom Kunden verwalteten Schlüssel direkt zu verwenden. Es beinhaltet Berechtigungen zur Ausführung des AWS KMS `Encrypt`, `Decrypt`, `ReEncrypt*`, `GenerateDataKey*`, und `DescribeKey` Operationen auf dem Schlüssel.
- Eine Anweisung, die die in der angegebene IAM-Identität zulässt `Principal` Element, das verwendet werden soll `CreateGrant` Vorgang zur Generierung von Zuschüssen, an die eine Teilmenge der eigenen Berechtigungen delegiert wird AWS-Services die integriert sind mit AWS KMS oder ein anderer Schulleiter. Auf diese Weise können sie den Schlüssel verwenden, um in Ihrem Namen verschlüsselte Ressourcen zu erstellen.

Ändern Sie keine vorhandenen Aussagen in der Police, wenn Sie die neuen Policen zu Ihrer wichtigsten Police hinzufügen.

Weitere Informationen finden Sie unter:

- Schlüssel erstellen in der AWS CLI Befehlsreferenz
- Put-Key-Richtlinie in der AWS CLI Befehlsreferenz
- Finden Sie die Schlüssel-ID und den Schlüssel-ARN in der AWS Key Management Service Leitfaden für Entwickler
- Service-linked Rollen für AWS 5 STÜCK
- Amazon EBS-Verschlüsselung in der Amazon EBS-Benutzerhandbuch
- AWS Key Management Service im AWS Key Management Service Leitfaden für Entwickler

Beispiel 1: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel erlauben

Fügen Sie der Schlüsselrichtlinie des vom Kunden verwalteten Schlüssels die folgenden Richtlinienerklärungen hinzu. Ersetzen Sie den Beispiel-ARN durch den ARN Ihres `AWSServiceRoleForPCS` dienstgebundene Rolle. Diese Beispielrichtlinie enthält AWS Mit dem PCS-Dienst verknüpfte Rolle (`AWSServiceRoleForPCS`) Berechtigungen zur Verwendung des vom Kunden verwalteten Schlüssels.

```
{
   "Sid": "Allow service-linked role use of the customer managed key",
   "Effect": "Allow",
   "Principal": {
       "AWS": [
           "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
       ]
   },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
   ],
   "Resource": "*"
}
```

```
{
   "Sid": "Allow attachment of persistent resources",
```

```
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::account-id:role/aws-service-role/pcs.amazonaws.com/
AWSServiceRoleForPCS"
        ]
    },
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": true
        }
     }
}
```

Beispiel 2: Schlüsselrichtlinienabschnitte, welche Zugriff auf den kundenverwalteten Schlüssel über mehrere Konten erlauben

Wenn Sie einen vom Kunden verwalteten Schlüssel in einem anderen Konto als Ihrem erstellen AWS PCS-Cluster, Sie müssen einen verwenden Zuschuss in Kombination mit der Schlüsselrichtlinie, um den kontoübergreifenden Zugriff auf den Schlüssel zu ermöglichen.

Um den Zugriff auf den Schlüssel zu gewähren

1. Fügen Sie der Schlüsselrichtlinie des vom Kunden verwalteten Schlüssels die folgenden Richtlinienerklärungen hinzu. Ersetzen Sie den Beispiel-ARN durch den ARN des anderen Kontos. Ersetzen *111122223333* mit der tatsächlichen Konto-ID des AWS-Konto dass Sie das erstellen möchten AWS PCS-Cluster in. Damit können Sie einem IAM-Benutzer oder einer IAM-Rolle im angegebenen Konto die Berechtigung erteilen, mit dem folgenden CLI-Befehl eine Berechtigung für den Schlüssel zu erstellen. Standardmäßig haben Benutzer keinen Zugriff auf den Schlüssel.

   ```
   {.
       "Sid": "Allow external account 111122223333 use of the customer managed key",
       "Effect": "Allow",
       "Principal": {
           "AWS": [
               "arn:aws:iam::111122223333:root"
   ```

```
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

```
{
    "Sid": "Allow attachment of persistent resources in external
 account 111122223333",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::111122223333:root"
        ]
    },
    "Action": [
        "kms:CreateGrant"
    ],
    "Resource": "*"
}
```

2. Von dem Konto aus, das Sie erstellen möchten AWS Erstellen Sie im PCS-Cluster einen Grant, der die entsprechenden Berechtigungen an den delegiert AWS Mit dem PCS-Dienst verknüpfte Rolle. Der Wert von `grantee-principal` ist der ARN der dienstverknüpften Rolle. Der Wert von `key-id` ist der ARN des Schlüssels.

   Das folgende Beispiel create-grant Der CLI-Befehl gibt der dienstverknüpften Rolle den Namen `AWSServiceRoleForPCS` im Konto *111122223333* Berechtigungen zur Verwendung des vom Kunden verwalteten Key-in-Kontos *444455556666*.

```
aws kms create-grant \
  --region us-west-2 \
  --key-id arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
  --grantee-principal arn:aws:iam::111122223333:role/aws-service-role/
pcs.amazonaws.com/AWSServiceRoleForPCS \
```

```
 --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
"GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"
```

> **Note**
>
> Der Benutzer, der die Anfrage stellt, muss über Berechtigungen zur Verwendung von verfügen `kms:CreateGrant` Aktion.

Das folgende Beispiel für eine IAM-Richtlinie ermöglicht eine IAM-Identität (Benutzer oder Rolle) im Konto *111122223333* um einen Zuschuss für das vom Kunden verwaltete Key-in-Konto zu gewähren *444455556666*.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreationOfGrantForTheKMSKeyinExternalAccount444455556666",
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
  ]
}
```

Weitere Informationen über die Erstellung eines Zuschusses für einen KMS-Schlüssel in einem anderen AWS-Konto, finden Sie unter Berechtigungserteilungen in AWS KMS im AWS Key Management Service -Entwicklerhandbuch.

> **Important**
>
> Der Name der serviceverknüpften Rolle, der als Prinzipal des Empfängers angegeben wird, muss der Name einer vorhandenen Rolle sein. Nachdem Sie den Zuschuss eingerichtet haben, um sicherzustellen, dass der Zuschuss Folgendes ermöglicht

AWS Wenn PCS den angegebenen KMS-Schlüssel verwenden soll, löschen Sie die dienstverknüpfte Rolle nicht und erstellen Sie sie erneut.

Bearbeiten Sie die wichtigsten Richtlinien in der AWS KMS Konsole

Die Beispiele in den vorherigen Abschnitten zeigen nur, wie einer Schlüsselrichtlinie Anweisungen hinzugefügt werden, was nur eine Möglichkeit darstellt, eine Schlüsselrichtlinie zu ändern. Der einfachste Weg, eine wichtige Richtlinie zu ändern, ist die Verwendung von AWS KMS die Standardansicht der Konsole für wichtige Richtlinien und machen Sie eine IAM-Identität (Benutzer oder Rolle) zu einer der wichtige Benutzer für die entsprechende Schlüsselrichtlinie. Weitere Informationen finden Sie unter Mit dem AWS-Managementkonsole Standardansicht in der AWS Key Management Service Leitfaden für Entwickler.

Warning

Die Standardansichtsrichtlinien der Konsole enthalten Berechtigungen zur Ausführung AWS KMS `Revoke` Operationen mit dem vom Kunden verwalteten Schlüssel. Wenn Sie einen Zuschuss widerrufen, der eine AWS-Konto Zugriff auf einen vom Kunden verwalteten Schlüssel in Ihrem Konto, Benutzer in diesem AWS-Konto verlieren den Zugriff auf die verschlüsselten Daten und den Schlüssel.

Zugriff AWS Parallel Computing Service mithilfe eines Schnittstellenendpunkts (AWS PrivateLink)

Du kannst verwenden AWS PrivateLink um eine private Verbindung zwischen Ihrer VPC und herzustellen AWS Parallel Computing Service (AWS PCS). Sie können darauf zugreifen AWS PCS als ob es in Ihrer VPC wäre, ohne die Verwendung eines Internet-Gateways, eines NAT-Geräts, einer VPN-Verbindung oder Direct Connect Verbindung. Instanzen in Ihrer VPC benötigen keine öffentlichen IP-Adressen für den Zugriff AWS PCS.

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 PCS bestimmt ist.

Weitere Informationen finden Sie unter Zugriff AWS-Services durch AWS PrivateLink in der AWS PrivateLink Führer.

Überlegungen für AWS PCS

Bevor Sie einen Schnittstellenendpunkt einrichten für AWS PCS, überprüfen Greifen Sie über einen Schnittstellen-VPC-Endpunkt auf einen AWS-Service zu in der AWS PrivateLink Führer.

AWS PCS unterstützt das Aufrufen aller seiner API-Aktionen über den Schnittstellenendpunkt.

⚠ Dual-stack Endpunktanforderung

Standardmäßig ist der AWS PCS Der Agent stellt eine Verbindung zum Dual-Stack-Endpunkt (IPv4 und IPv6) her `pcs.region.api.aws`. Dieser Name muss in die private IP-Adresse eines Schnittstellen-VPC-Endpunkts in Ihrer VPC aufgelöst werden. Ersetzen *`region`* mit der ID des AWS-Region, wie `us-east-1`.

- Wenn du das benutzt Amazon-provided DNS, behalte DNS-Namen aktivieren ausgewählt unter Zusätzliche Einstellungen wenn Sie den Endpunkt erstellen und sicherstellen, dass DNS-Hostnamen und DNS-Auflösung für die VPC aktiviert sind.
- Wenn Sie benutzerdefiniertes DNS verwenden, stellen Sie sicher, dass es aufgelöst werden kann `pcs.region.api.aws` zum Endpunkt.

Wenn Ihre VPC keinen direkten Internetzugang hat, müssen Sie einen VPC-Endpunkt konfigurieren, damit Ihre Compute-Knotengruppen-Instances den AWS PCS `RegisterComputeNodeGroupInstance` API-Aktion.

Erstellen Sie einen Schnittstellenendpunkt für AWS PCS

Sie können einen Schnittstellenendpunkt erstellen für AWS PCS entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface (AWS CLI). Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für AWS PCS mit dem folgenden Dienstnamen:

```
com.amazonaws.region.pcs
```

Ersetzen *`region`* mit der ID des AWS-Region um den Endpunkt in zu erstellen, wie zum Beispiel `us-east-1`.

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen stellen an AWS PCS unter Verwendung seines regionalen DNS-Namens. Beispiel, `pcs.us-east-1.api.aws`.

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 vollen Zugriff auf AWS PCS über den Schnittstellenendpunkt. Um den erlaubten Zugriff zu kontrollieren AWS PCS Hängen Sie von Ihrer VPC aus eine benutzerdefinierte Endpunktrichtlinie an den Schnittstellenendpunkt an.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Prinzipale, die Aktionen ausführen können (AWS-Konten, IAM-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 AWS PCS actions

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Endpunktrichtlinie. Wenn Sie diese Richtlinie an Ihren Schnittstellenendpunkt anhängen, gewährt sie Zugriff auf die aufgelisteten AWS PCS Aktionen für alle Prinzipale des Clusters mit den angegebenen *`cluster-id`*. Ersetzen *`region`* mit der ID des AWS-Region des Clusters, wie `us-east-1`. Ersetzen *`account-id`* mit dem AWS-Konto Nummer des Clusters.

```
{
    "Statement": [
            {
                "Action": [
                "pcs:CreateCluster",
                "pcs:ListClusters",
                "pcs:DeleteCluster",
                "pcs:GetCluster",
                ],
                "Effect": "Allow",
                "Principal": "*",
                "Resource": [
```

```
                    "arn:aws:pcs:region:account-id:cluster/cluster-id*"
                ]
            }
        ]
}
```

Identitäts- und Zugriffsmanagement für AWS Dienst für parallele Datenverarbeitung

AWS Identity and Access Management (IAM) ist ein AWS-Service das hilft einem Administrator, den Zugriff auf sicher zu kontrollieren AWS Ressourcen. IAM-Administratoren kontrollieren, wer sein kann authentifiziert (angemeldet) und autorisiert (über Berechtigungen verfügen) zur Nutzung AWS PCS-Ressourcen. IAM ist ein AWS-Service die Sie ohne zusätzliche Kosten nutzen können.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- Wie AWS Parallel Computing Service arbeitet mit IAM
- Identity-based politische Beispiele für AWS Dienst für parallele Datenverarbeitung
- AWS verwaltete Richtlinien für AWS Dienst für parallele Datenverarbeitung
- Service-linked Rollen für AWS 5 STÜCK
- Amazon EC2 Spot-Rolle für AWS 2 STCK
- Mindestberechtigungen für AWS 5 STÜCK
- IAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung
- Fehlerbehebung AWS Identität und Zugriff auf den Parallel Computing Service

Zielgruppe

Wie benutzt du AWS Identity and Access Management (IAM) unterscheidet sich je nach Ihrer Rolle:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung AWS Identität und Zugriff auf den Parallel Computing Service).

- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Wie AWS Parallel Computing Service arbeitet mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe Identity-based politische Beispiele für AWS Dienst für parallele Datenverarbeitung).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich anmelden AWS mit Ihren Identitätsdaten. Sie müssen authentifiziert sein als Root-Benutzer des AWS-Kontos, ein IAM-Benutzer oder indem Sie eine IAM-Rolle übernehmen.

Sie können sich als Verbundidentität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single-Sign-On-Authentifizierung oder Google/Facebook Anmeldeinformationen. Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

Für den programmatischen Zugriff AWS bietet ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie eine erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität namens AWS-Konto Root-Benutzer das hat vollen Zugriff auf alle AWS-Services und Ressourcen. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, menschliche Benutzer dazu zu verpflichten, einen Verbund mit einem Identitätsanbieter für den Zugriff zu verwenden AWS-Services Verwendung temporärer Anmeldeinformationen.

EIN föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Web Identity Provider oder Directory Service das zugreift AWS-Services unter Verwendung von Anmeldeinformationen aus einer Identitätsquelle. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie unter Erfordern Sie, dass menschliche Benutzer einen Verbund mit einem Identitätsanbieter für den Zugriff verwenden AWS Verwendung temporärer Anmeldeinformationen in der IAM-Benutzerhandbuch.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer- zu einer IAM-Rolle wechseln (Konsole) oder indem Sie einen anrufen AWS CLI oder AWS API-Betrieb. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff in AWS indem Sie Richtlinien erstellen und sie anhängen AWS Identitäten oder Ressourcen. Eine Richtlinie definiert Berechtigungen, wenn sie einer Identität oder Ressource zugeordnet sind. AWS wertet diese Richtlinien aus, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien sind gespeichert in AWS als JSON-Dokumente. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Politik

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten 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.

Identity-based Richtlinien können wie folgt sein integrierte Richtlinien (direkt in eine einzige Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind). Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Resource-based Politik

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen 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. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Resource-based Richtlinien sind Inline-Richtlinien, die sich in diesem Dienst befinden. Sie können nicht verwenden AWS verwaltete Richtlinien von IAM in einer ressourcenbasierten Richtlinie.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximale Anzahl von Berechtigungen festgelegt werden kann, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Service-Kontrollrichtlinien (SCPs) – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.

- Ressourcen-Kontrollrichtlinien (RCPs) – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter Ressourcen-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die daraus resultierenden Berechtigungen komplizierter zu verstehen. Um zu erfahren, wie AWS bestimmt, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, siehe Logik der Bewertung von Richtlinien in der IAM-Benutzerhandbuch.

Wie AWS Parallel Computing Service arbeitet mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf zu verwalten AWS PCS, erfahren Sie, mit welchen IAM-Funktionen Sie verwenden können AWS STCK.

IAM-Funktionen, die Sie verwenden können AWS Dienst für parallele Datenverarbeitung

IAM-Feature	AWS PCS-Unterstützung
Identity-based Politik	Ja
Resource-based Politik	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja

IAM-Feature	AWS PCS-Unterstützung
Prinzipalberechtigungen	Ja
Servicerollen	Nein
Service-linked Rollen	Ja

Um einen allgemeinen Überblick darüber zu erhalten, wie AWS PCS und andere AWS Dienste funktionieren mit den meisten IAM-Funktionen, siehe AWS Dienste, die mit IAM funktionieren in der IAM-Benutzerhandbuch.

Identity-based Richtlinien für AWS STCK

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität anhängen können, z. B. an einen IAM-Benutzer, eine Benutzergruppe oder eine Rolle. 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. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Identity-based Richtlinienbeispiele für AWS STCK

Um Beispiele anzusehen von AWS Auf der Identität basierende PCS-Richtlinien finden Sie unter Identity-based politische Beispiele für AWS Dienst für parallele Datenverarbeitung.

Resource-based Richtlinien innerhalb AWS STCK

Unterstützt ressourcenbasierte Richtlinien: Nein

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. 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. Principals können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Politische Maßnahmen für AWS 5 STCK

Unterstützt Richtlinienaktionen: Ja

Administratoren können verwenden AWS JSON-Richtlinien, um festzulegen, 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. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu gewähren.

Um eine Liste von zu sehen AWS PCS-Aktionen finden Sie unter Aktionen, definiert durch AWS Dienst für parallele Datenverarbeitung in der Referenz zur Serviceautorisierung.

Politische Maßnahmen in AWS PCS verwendet vor der Aktion das folgende Präfix:

```
pcs
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "pcs:action1",
      "pcs:action2"
         ]
```

Politische Ressourcen für AWS STCK

Unterstützt Richtlinienressourcen: Ja

Administratoren können verwenden AWS JSON-Richtlinien, um festzulegen, 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. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Um eine Liste von zu sehen AWS PCS-Ressourcentypen und ihre ARNs finden Sie unter Ressourcen, definiert durch AWS Dienst für parallele Datenverarbeitung in der Referenz zur Serviceautorisierung. Informationen zu den Aktionen, mit denen Sie den ARN der einzelnen Ressourcen angeben können, finden Sie unter Aktionen, definiert durch AWS Dienst für parallele Datenverarbeitung .

Um Beispiele zu sehen von AWS Auf der Identität basierende PCS-Richtlinien finden Sie unter Identity-based politische Beispiele für AWS Dienst für parallele Datenverarbeitung.

Schlüssel zu den Bedingungen der Richtlinie für AWS 5 STÜCK

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können verwenden AWS JSON-Richtlinien, um festzulegen, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. 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. Um alles zu sehen AWS globale Bedingungsschlüssel finden Sie unter AWS globale Bedingungskontextschlüssel in der IAM-Benutzerhandbuch.

Um eine Liste von zu sehen AWS PCS-Bedingungsschlüssel finden Sie unter Konditionsschlüssel für AWS Dienst für parallele Datenverarbeitung in der Referenz zur Serviceautorisierung. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Aktionen Definiert von AWS Dienst für parallele Datenverarbeitung .

Um Beispiele zu sehen von AWS Auf der Identität basierende PCS-Richtlinien finden Sie unter Identity-based politische Beispiele für AWS Dienst für parallele Datenverarbeitung.

ACLs rein AWS STCK

Unterstützt ACLs: Nein

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit AWS 5 STCK

Unterstützt ABAC (Tags in Richtlinien): Ja

Attribute-based Access Control (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert, die als Tags bezeichnet werden. Sie können IAM-Entitäten Tags anhängen und AWS Ressourcen und entwerfen Sie dann ABAC-Richtlinien, um Operationen zu ermöglichen, wenn das Tag des Principals mit dem Tag auf der Ressource übereinstimmt.

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 STCK

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie den Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu

verwenden. Weitere Informationen finden Sie unter Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren im IAM-Benutzerhandbuch.

Cross-service Hauptberechtigungen für AWS STCK

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Rechte des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung AWS-Service um Anfragen an nachgelagerte Dienste zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AWS STCK

Unterstützt Servicerollen: Nein

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**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann zu Problemen führen AWS PCS-Funktionalität. Bearbeiten Sie Servicerollen nur, wenn AWS PCS bietet dazu Anleitungen.

Service-linked Rollen für AWS 5 STÜCK

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstgebundene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist AWS-Service. Der Dienst kann die Rolle übernehmen, eine Aktion in Ihrem Namen auszuführen. Service-linked Rollen erscheinen in Ihrem AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Weitere Informationen zum Erstellen oder Verwalten AWS Mit dem PCS-Dienst verknüpfte Rollen finden Sie unter Service-linked Rollen für AWS 5 STÜCK.

Identity-based politische Beispiele für AWS Dienst für parallele Datenverarbeitung

Standardmäßig sind Benutzer und Rollen nicht berechtigt, etwas zu erstellen oder zu ändern AWS PCS-Ressourcen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen.

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.

Für Details zu Aktionen und Ressourcentypen, definiert durch AWS PCS, einschließlich des Formats der ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS Dienst für parallele Datenverarbeitung in der Referenz zur Serviceautorisierung.

Themen

- Best Practices für Richtlinien
- Verwendung der AWS PCS-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand etwas erstellen, darauf zugreifen oder löschen kann AWS PCS-Ressourcen in Ihrem Konto. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Wenn Sie identitätsbasierte Richtlinien erstellen oder bearbeiten, folgen Sie diesen Richtlinien und Empfehlungen:

- Beginnen Sie mit AWS verwaltete Richtlinien und Umstellung auf Berechtigungen mit geringsten Rechten — Um mit der Vergabe von Berechtigungen für Ihre Benutzer und Workloads zu beginnen, verwenden Sie AWS verwaltete Richtlinien die Berechtigungen für viele gängige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen, die Berechtigungen weiter zu reduzieren, indem Sie Folgendes definieren AWS vom Kunden verwaltete Richtlinien, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien oder Von AWS verwaltete Richtlinien für Auftragsfunktionen 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 im Rahmen einer bestimmten Aktion genutzt werden AWS-Service, wie CloudFormation. 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.
- Mehrstufige Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem AWS-Konto, schalten Sie MFA für zusätzliche Sicherheit ein. 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 Best Practices für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwendung der AWS PCS-Konsole

Um auf die zuzugreifen AWS Für die Parallel Computing Service Console müssen Sie über ein Mindestmaß an Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS PCS-Ressourcen in Ihrem 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 Anrufe tätigen, keine Mindestberechtigungen für die Konsole gewähren AWS CLI oder der AWS API. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Weitere Informationen zu den Mindestberechtigungen, die für die Nutzung der erforderlich sind AWS PCS-Konsole finden Sie unter Mindestberechtigungen für AWS 5 STÜCK.

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 oder AWS 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 Dienst für parallele Datenverarbeitung

Ein AWS Eine verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Denken Sie daran AWS verwaltete Richtlinien gewähren möglicherweise keine Berechtigungen mit geringsten Rechten für Ihre speziellen Anwendungsfälle, da sie für alle verfügbar sind AWS zu verwendende Kunden. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die unter definierten Berechtigungen nicht ändern AWS verwaltete Richtlinien. Wenn AWS aktualisiert die in einem definierten Berechtigungen AWS verwaltete Richtlinie. Das Update wirkt sich auf alle Hauptidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am ehesten eine AWS verwaltete Richtlinie, wenn eine neue AWS-Service wird gestartet oder neue API-Operationen werden für bestehende Dienste verfügbar.

Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AWSPCSComputeNodePolicy

Sie können anhängen AWSPCSComputeNodePolicy an Ihre IAM-Entitäten. Sie können diese Richtlinie an eine anhängen AWS Die IAM-Rolle für PCS-Rechenknoten, die Sie angeben, damit Knoten, die diese Rolle verwenden, eine Verbindung zu einem herstellen können AWS PCS-Cluster.

AWS PCS hängt diese Richtlinie an eine Rechenknotengruppenrolle an, wenn Sie die Konsole verwenden, um eine Rechenknotengruppe zu erstellen.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `pcs:RegisterComputeNodeGroupInstance` — Erlaube eine AWS PCS-Rechenknoten (EC2-Instanz) zur Registrierung bei einem AWS PCS-Cluster.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSPCSComputeNodePolicy in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSPCSServiceRolePolicy

Sie können nicht anhängen AWSPCSServiceRolePolicy zu Ihren IAM-Entitäten. Diese Richtlinie ist an eine dienstgebundene Rolle angehängt, die Folgendes ermöglicht AWS PCS, um Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Service-linked Rollen für AWS 5 STÜCK.

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen.

- `ec2` — Erlaubt AWS PCS zum Erstellen und Verwalten von Amazon EC2-Ressourcen.
- `iam` — Erlaubt AWS PCS, eine serviceverknüpfte Rolle für die Amazon EC2-Flotte zu erstellen und die Rolle an Amazon EC2 zu übergeben.
- `cloudwatch` — Erlaubt AWS PCS, Servicemetriken auf Amazon zu veröffentlichen CloudWatch.
- `secretsmanager` — Erlaubt AWS PCS zur Verwaltung von Geheimnissen für AWS PCS-Cluster-Ressourcen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter AWSPCSServiceRolePolicy in der Referenz zu von AWS verwalteten Richtlinien.

AWS PCS aktualisiert auf AWS Verwaltete Richtlinien

Sehen Sie sich Details zu Updates an AWS verwaltete Richtlinien für AWS PCS, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen. Für automatische Benachrichtigungen über Änderungen an dieser Seite abonnieren Sie den RSS-Feed auf der AWS Seite mit dem Verlauf des PCS-Dokuments.

Änderungen	Beschreibung	Datum
AWSPCSServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	AWS PCS hat neue Berechtigungen hinzugefügt, um mehrere Netzwerkschnittstellen für Recheninstanzen zu unterstützen. Hinzugefügt `ec2:secondary-interface` und	30. September 2026

Änderungen	Beschreibung	Datum
	`ec2:secondary-subnet` zuzulassende Ressourcentypen AWS PCS, um diese Ressourcen zu erstellen und zu verwalten.	
AWSPCSServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	AWS PCS hat neue Berechtigungen zur Unterstützung von Kapazitätsblöcken hinzugefügt, um eine vorhersehbare Rechenkapazität zu gewährleisten. Hinzugefügt `ec2:DescribeCapacityReservations` Erlaubnis zum Aktivieren AWS PCS, um Capacity Block-Reservierungen für Rechenknotengruppen zu erkennen und zu verwenden.	11. September 2025
AWSPCSComputeNodePolicy – Neue Richtlinie	AWS PCS hat eine neue Richtlinie hinzugefügt, um folgende Berechtigungen zu erteilen AWS PCS-Rechenknoten, zu denen eine Verbindung hergestellt werden soll AWS PCS-Cluster. AWS PCS hängt diese Richtlinie an eine IAM-Rolle an, wenn Sie eine Compute-Knotengruppe in der AWS PCS-Konsole.	23. Juni 2025

Änderungen	Beschreibung	Datum
Die JSON-Datei in diesem Dokument wurde aktualisiert	Der JSON-Code in diesem Dokument wurde korrigiert und enthält nun `"arn:aws:ec2:*:*:spot-instances-request/*"` .	5. September 2024
AWS PCS hat begonnen, Änderungen zu verfolgen	AWS PCS begann mit der Nachverfolgung der Änderungen für AWS verwaltete Richtlinien.	28. August 2024

Service-linked Rollen für AWS 5 STÜCK

AWS Der Parallel Computing Service verwendet AWS Identity and Access Management (ICH BIN) dienstgebundene Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger IAM-Rollentyp, der direkt mit verknüpft ist AWS STCK. Service-linked Rollen sind vordefiniert durch AWS PCS und enthalten alle Berechtigungen, die der Dienst benötigt, um andere Dienste aufzurufen AWS Dienstleistungen in Ihrem Namen.

Eine dienstbezogene Rolle macht die Einrichtung AWS PCS ist einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS PCS definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, nur AWS PCS kann seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen juristischen Stelle von IAM zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Das schützt Ihre AWS PCS-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Diensten, die dienstbezogene Rollen unterstützen, finden Sie unter AWS Dienste, die mit IAM funktionieren und suchen Sie nach den Diensten, die Ja in der Service-linked Rollen Spalte. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Service-linked Rollenberechtigungen für AWS STCK

AWS PCS verwendet die dienstverknüpfte Rolle mit dem Namen AWSServiceRoleForPCS — Erteilt die Erlaubnis für AWS PCS zur Verwaltung von Amazon EC2-Ressourcen.

Das AWSServiceRoleForPCS Die dienstgebundene Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `pcs.amazonaws.com`

Die Richtlinie für Rollenberechtigungen heißt AWSPCSServiceRolePolicy erlaubt AWS PCS, Aktionen für bestimmte Ressourcen abzuschließen.

Sie müssen Berechtigungen konfigurieren, damit eine Benutzer, Gruppen oder Rollen eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen können. Weitere Informationen finden Sie unter Service-linked Rollenberechtigungen in der IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für AWS STCK.

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. AWS PCS erstellt eine serviceverknüpfte Rolle für Sie, wenn Sie einen Cluster erstellen.

Bearbeiten einer serviceverknüpften Rolle für AWS PCS

AWS PCS erlaubt Ihnen nicht, das zu bearbeiten AWSServiceRoleForPCS serviceverknüpfte Rolle. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für AWS STCK.

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der AWS Der PCS-Dienst verwendet die Rolle, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um zu entfernen AWS PCS-Ressourcen, die von der verwendet werden AWSServiceRoleForPCS

Sie müssen alle Ihre Cluster löschen, um das zu löschen AWSServiceRoleForPCS serviceverknüpfte Rolle. Weitere Informationen finden Sie unter Löschen Sie einen Cluster.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die AWS CLI, oder die AWS API zum Löschen der AWSServiceRoleForPCS Mit einem Dienst verknüpfte Rolle. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Unterstützte Regionen für AWS Mit dem PCS-Dienst verknüpfte Rollen

AWS PCS unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS -Regionen und Endpunkte.

Amazon EC2 Spot-Rolle für AWS 2 STCK

Wenn Sie eine erstellen möchten AWS PCS-Rechenknotengruppe, die verwendet Spot als Kaufoption müssen Sie auch die haben AWSServiceRoleForEC2Spot servicebasierte Rolle in Ihrem AWS-Konto. Sie können Folgendes verwenden AWS CLI Befehl zum Erstellen der Rolle. Weitere Informationen finden Sie unter Erstellen Sie eine serviceverknüpfte Rolle und Erstellen Sie eine Rolle, um Berechtigungen an einen zu delegieren AWS Bedienung in der AWS Identity and Access Management Benutzerleitfaden.

```
aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

> **Note**
>
> Sie erhalten den folgenden Fehler, wenn AWS-Konto hat bereits eine `AWSServiceRoleForEC2Spot` IAM-Rolle.

```
 An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
operation: Service role name AWSServiceRoleForEC2Spot has been taken in this
account, please try a different suffix.
```

Mindestberechtigungen für AWS 5 STÜCK

In diesem Abschnitt werden die IAM-Berechtigungen beschrieben, die für eine IAM-Identität (Benutzer, Gruppe oder Rolle) zur Nutzung des Dienstes erforderlich sind.

Inhalt

- Mindestberechtigungen zur Verwendung von API-Aktionen
- Mindestberechtigungen zur Verwendung von Stichwörtern
- Mindestberechtigungen zur Unterstützung von Protokollen
- Mindestberechtigungen zur Verwendung von Capacity Blocks
- Mindestberechtigungen für einen Dienstadministrator

Mindestberechtigungen zur Verwendung von API-Aktionen

API-Aktion	Mindestberechtigungen	Zusätzliche Berechtigungen für die Konsole
CreateCluster	`ec2:CreateNetworkInterface, ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:GetSecurityGroupsForVpc, iam:CreateServiceLinkedRole, secretsmanager:CreateSecret, secretsmanager:TagResource,`	

API-Aktion	Mindestberechtigungen	Zusätzliche Berechtigungen für die Konsole
	`secretsmanager:RotateSecret`, `pcs:CreateCluster`	
ListClusters	`pcs:ListClusters`	
GetCluster	`pcs:GetCluster`	`ec2:DescribeSubnets`
DeleteCluster	`pcs:DeleteCluster`	
CreateComputeNodeGroup	`ec2:DescribeVpcs`, `ec2:DescribeSubnets`, `ec2:DescribeSecurityGroups`, `ec2:DescribeLaunchTemplates`, `ec2:DescribeLaunchTemplateVersions`, `ec2:DescribeInstanceTypes`, `ec2:DescribeInstanceTypeOfferings`, `ec2:RunInstances`, `ec2:CreateFleet`, `ec2:CreateTags`, `iam:PassRole`, `iam:GetInstanceProfile`, `pcs:CreateComputeNodeGroup`	`iam:ListInstanceProfiles`, `ec2:DescribeImages`, `pcs:GetCluster`
ListComputerNodeGroups	`pcs:ListComputeNodeGroups`	`pcs:GetCluster`

API-Aktion	Mindestberechtigungen	Zusätzliche Berechtigungen für die Konsole
GetComputeNodeGroup	`pcs:GetComputeNodeGroup`	`ec2:DescribeSubnets`
UpdateComputeNodeGroup	`ec2:DescribeVpcs, ec2:DescribeSubnets, ec2:DescribeSecurityGroups, ec2:DescribeLaunchTemplates, ec2:DescribeLaunchTemplateVersions, ec2:DescribeInstanceTypes, ec2:DescribeInstanceTypeOfferings, ec2:RunInstances, ec2:CreateFleet, ec2:CreateTags, iam:PassRole, iam:GetInstanceProfile, pcs:UpdateComputeNodeGroup`	`pcs:GetComputeNodeGroup, iam:ListInstanceProfiles, ec2:DescribeImages, pcs:GetCluster`
DeleteComputeNodeGroup	`pcs:DeleteComputeNodeGroup`	
CreateQueue	`pcs:CreateQueue`	`pcs:ListComputeNodeGroups, pcs:GetCluster`
ListQueues	`pcs:ListQueues`	`pcs:GetCluster`
GetQueue	`pcs:GetQueue`	

API-Aktion	Mindestberechtigungen	Zusätzliche Berechtigungen für die Konsole
UpdateQueue	`pcs:UpdateQueue`	`pcs:ListComputeNodeGroups, pcs:GetQueue`
DeleteQueue	`pcs:DeleteQueue`	

Mindestberechtigungen zur Verwendung von Stichwörtern

Die folgenden Berechtigungen sind erforderlich, um Tags mit Ihren Ressourcen in verwenden zu können AWS STCK.

```
pcs:ListTagsForResource,
pcs:TagResource,
pcs:UntagResource
```

Mindestberechtigungen zur Unterstützung von Protokollen

AWS PCS sendet Protokolldaten an Amazon CloudWatch Protokolle (CloudWatch Protokolle). Sie müssen sicherstellen, dass Ihre Identität über die Mindestberechtigungen zur Verwendung verfügt CloudWatch Protokolle. Weitere Informationen finden Sie unter Überblick über die Verwaltung der Zugriffsberechtigungen für CloudWatch Protokolliert Ressourcen in der Amazonas CloudWatch Logs — Benutzerhandbuch.

Für Informationen zu den Berechtigungen, die ein Dienst benötigt, um Protokolle zu senden CloudWatch Protokolle finden Sie unter Protokollierung wird aktiviert von AWS Dienstleistungen in der Amazonas CloudWatch Logs — Benutzerhandbuch.

Mindestberechtigungen zur Verwendung von Capacity Blocks

Amazon EC2 Capacity Blocks for ML ist eine Amazon EC2-Kaufoption, mit der Sie für Reservierungen im Voraus bezahlen können GPU-based beschleunigte Recheninstanzen innerhalb eines bestimmten Datums- und Zeitbereichs zur Unterstützung kurzzeitiger Workloads. Weitere Informationen finden Sie unter Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit AWS 2 STÜCK.

Sie entscheiden sich für die Verwendung von Kapazitätsblöcken, wenn Sie eine Rechenknotengruppe erstellen oder aktualisieren. Die IAM-Identität, die Sie zum Erstellen oder Aktualisieren der Compute-Knotengruppe verwenden, muss über die folgenden Berechtigungen verfügen:

```
ec2:DescribeCapacityReservations,
ec2:DescribeCapacityBlocks,
ec2:DescribeCapacityBlockStatus
```

Mindestberechtigungen für einen Dienstadministrator

Die folgende IAM-Richtlinie legt die Mindestberechtigungen fest, die für eine IAM-Identität (Benutzer, Gruppe oder Rolle) erforderlich sind, um die zu konfigurieren und zu verwalten AWS PCS-Dienst.

> **Note**
>
> Benutzer, die den Dienst nicht konfigurieren und verwalten, benötigen diese Berechtigungen nicht. Benutzer, die nur Jobs ausführen, verwenden Secure Shell (SSH), um eine Verbindung zum Cluster herzustellen. AWS Identity and Access Management (IAM) kümmert sich nicht um die Authentifizierung oder Autorisierung für SSH.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PCSAccess",
      "Effect": "Allow",
      "Action": [
        "pcs:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2Access",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeImages",
        "ec2:GetSecurityGroupsForVpc",
        "ec2:DescribeSubnets",
```

```
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:RunInstances",
        "ec2:CreateFleet",
        "ec2:CreateTags",
        "ec2:DescribeCapacityReservations",
        "ec2:DescribeCapacityBlocks",
        "ec2:DescribeCapacityBlockStatus"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamInstanceProfile",
      "Effect": "Allow",
      "Action": [
        "iam:GetInstanceProfile"
      ],
      "Resource": "*"
    },
    {
      "Sid": "IamPassRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/*/AWSPCS*",
        "arn:aws:iam::*:role/AWSPCS*",
        "arn:aws:iam::*:role/aws-pcs/*",
        "arn:aws:iam::*:role/*/aws-pcs/*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ec2.amazonaws.com"
          ]
        }
      }
    },
    {
```

```
      "Sid": "SLRAccess",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/pcs.amazonaws.com/AWSServiceRoleFor*",
        "arn:aws:iam::*:role/aws-service-role/spot.amazonaws.com/AWSServiceRoleFor*"
      ],
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": [
            "pcs.amazonaws.com",
            "spot.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AccessKMSKey",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey",
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    },
    {
      "Sid": "SecretManagementAccess",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret",
        "secretsmanager:TagResource",
        "secretsmanager:UpdateSecret",
        "secretsmanager:RotateSecret"
      ],
      "Resource": "*"
    },
    {
       "Sid": "ServiceLogsDelivery",
       "Effect": "Allow",
```

```
        "Action": [
          "pcs:AllowVendedLogDeliveryForResource",
          "logs:PutDeliverySource",
          "logs:PutDeliveryDestination",
          "logs:CreateDelivery"
        ],
        "Resource": "*"
    }
  ]
}
```

IAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung

Zu den Anwendungen, die auf einer EC2-Instance ausgeführt werden, gehören AWS Anmeldeinformationen in beliebigen AWS API-Anfragen, die sie stellen. Wir empfehlen Ihnen, eine IAM-Rolle zu verwenden, um temporäre Anmeldeinformationen auf der EC2-Instance zu verwalten. Zu diesem Zweck können Sie ein Instance-Profil definieren und es an Ihre Instances anhängen. Weitere Informationen finden Sie unter IAM-Rollen für Amazon EC2 in der Amazon Elastic Compute Cloud-Benutzerhandbuch.

> **Note**
>
> Wenn du den benutzt AWS-Managementkonsole Um eine IAM-Rolle für Amazon EC2 zu erstellen, erstellt die Konsole automatisch ein Instance-Profil und gibt ihm denselben Namen wie die IAM-Rolle. Wenn Sie das verwenden AWS CLI, AWS API-Aktionen oder ein AWS SDK Um die IAM-Rolle zu erstellen, erstellen Sie das Instanzprofil als separate Aktion. Weitere Informationen finden Sie unter Instanzprofile in der Amazon Elastic Compute Cloud-Benutzerhandbuch.

Sie müssen den Amazon-Ressourcennamen (ARN) eines Instanzprofils angeben, wenn Sie eine Rechenknotengruppe erstellen. Sie können verschiedene Instanzprofile für einige oder alle Compute-Knotengruppen auswählen.

Voraussetzungen

IAM-Rolle des Instanzprofils

Die dem Instanzprofil zugeordnete IAM-Rolle muss `/aws-pcs/` in ihrem Pfad, oder ihr Name muss mit beginnen `AWSPCS`.

Beispiel für ARNs für IAM-Rollen

- `arn:aws:iam::*:role/AWSPCS-example-role-1`
- `arn:aws:iam::*:role/aws-pcs/example-role-2`

Berechtigungen

Die IAM-Rolle, die dem Instanzprofil für zugeordnet ist AWS PCS muss die folgende Richtlinie enthalten.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "pcs:RegisterComputeNodeGroupInstance"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Zusätzliche Richtlinien

Erwägen Sie, dem Instanzprofil verwaltete Richtlinien hinzuzufügen. Beispiel:

- AmazonS3ReadOnlyAccess bietet schreibgeschützten Zugriff auf alle S3-Buckets.
- AmazonSSMManagedInstanceCore ermöglicht die Kernfunktionen des AWS Systems Manager-Service, wie z. B. den Fernzugriff direkt von der Amazon Management Console aus.
- CloudWatchAgentServerPolicy enthält die für die Verwendung erforderlichen Berechtigungen AmazonCloudWatchAgent auf Servern.

Sie können auch Ihre eigenen IAM-Richtlinien hinzufügen, die Ihren speziellen Anwendungsfall unterstützen.

Erstellen Sie ein Instanzprofil für AWS STCK

AWS PCS console

Wählen Sie Erstellen Sie ein Basisprofil wenn Sie eine Compute-Knotengruppe erstellen, die Sie haben möchten AWS PCS erstellt eine für Sie mit der erforderlichen Mindestrichtlinie.

Amazon EC2 console

Sie können ein Instanzprofil direkt von der Amazon EC2-Konsole aus erstellen. Weitere Informationen finden Sie unter Verwenden von Instanzprofilen in der AWS Identity and Access Management Benutzerleitfaden.

> **Important**
>
> Stellen Sie sicher, dass Sie das erforderliche Präfix verwenden `AWSPCS` im IAM-Rollennamen.

AWS CLI

Einrichtung des Basic-Instanzprofils mithilfe der AWS-CLI

> **Note**
>
> Ersetzen *`example-role`* in den folgenden Beispielen durch den Namen Ihrer IAM-Rolle.

1. Erstellen Sie eine IAM-Rolle mit `/aws-pcs/` als Pfadattribut oder als Name, der beginnt mit `AWSPCS`.

   a. Kopieren Sie den folgenden Inhalt und fügen Sie ihn in eine neue Textdatei mit dem Namen ein `trust_policy.json`.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Principal": {
```

```
                "Service": [
                    "ec2.amazonaws.com"
                ]
            },
            "Action": [
                "sts:AssumeRole"
            ]
        }
    ]
}
```

b. Verwenden Sie einen der folgenden Befehle, um die IAM-Rolle zu erstellen.

```
aws iam create-role --path /aws-pcs/ --role-name example-role --assume-role-policy-document file://trust_policy.json
```

oder

```
aws iam create-role --role-name AWSPCS-example-role --assume-role-policy-document file://trust_policy.json
```

2. Fügen Sie Berechtigungen hinzu.

a. Kopieren Sie den folgenden Inhalt und fügen Sie ihn in eine neue Textdatei mit dem Namen ein `policy_document.json`.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "pcs:RegisterComputeNodeGroupInstance"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

b. Hängen Sie das Richtliniendokument an die Rolle an. Mit diesem Befehl wird die Richtlinie als integrierte Richtlinie angehängt.

```
aws iam put-role-policy \
    --role-name example-role \
    --policy-name pcsRegisterInstancePolicy \
    --policy-document file://policy_document.json
```

3. Erstellen Sie ein Instanzprofil. Ersetzen ***example-profile*** mit dem Namen Ihres Instanzprofils.

```
aws iam create-instance-profile --instance-profile-name example-profile
```

4. Ordnen Sie die IAM-Rolle dem Instanzprofil zu.

```
aws iam add-role-to-instance-profile \
   --instance-profile-name example-profile \
   --role-name example-role
```

Finden Sie die Instanzprofile, die verwendet werden mit AWS STCK

1. Wenn Sie die genauen Namen Ihrer IAM-Rollen für nicht kennen AWS PCS, verwenden Sie Folgendes AWS CLI Befehl zum Auflisten der IAM-Rollen, die den Anforderungen entsprechen AWS Anforderungen an den PCS-Namen.

```
aws iam list-roles --query "Roles[?starts_with(RoleName, 'AWSPCS') ||
 contains(Path, '/aws-pcs/')].[RoleName]" --output text
```

2. Verwenden Sie Folgendes AWS CLI Befehl, um die Instanzprofile aufzulisten, die einer bestimmten IAM-Rolle zugeordnet sind. Ersetzen *role-name* mit dem Namen einer IAM-Rolle, die erfüllt ist AWS Anforderungen an den PCS-Namen.

```
aws iam list-instance-profiles-for-role --role-name role-name
```

Fehlerbehebung AWS Identität und Zugriff auf den Parallel Computing Service

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS PCS und IAM.

Themen

- Ich bin nicht berechtigt, eine Aktion auszuführen in AWS 5 STCK
- Ich bin nicht berechtigt, iam: auszuführenPassRole
- Ich möchte Leute außerhalb von mir zulassen AWS-Konto um auf meine zuzugreifen AWS PCS-Ressourcen

Ich bin nicht berechtigt, eine Aktion auszuführen in AWS 5 STCK

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven *`my-example-widget`*-Ressource anzuzeigen, jedoch nicht über `pcs:`*`GetWidget`*-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 pcs:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `pcs:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam: auszuführenPassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, das auszuführen `iam:PassRole` Aktion, Ihre Richtlinien müssen aktualisiert werden, damit Sie eine Rolle übergeben können AWS 5 STCK.

Einige AWS-Services ermöglichen es Ihnen, eine vorhandene Rolle an diesen Dienst zu übergeben, anstatt eine neue Dienstrolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer den Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion auszuführen in AWS STCK. 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 Leute außerhalb von mir zulassen AWS-Konto um auf meine zuzugreifen AWS PCS-Ressourcen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Um zu erfahren, ob AWS PCS unterstützt diese Funktionen, siehe Wie AWS Parallel Computing Service arbeitet mit IAM.
- Um zu erfahren, wie Sie überall Zugriff auf Ihre Ressourcen gewähren AWS-Konten die Sie besitzen, finden Sie Bereitstellung des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto das Sie besitzen in der IAM-Benutzerhandbuch.
- Um zu erfahren, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren AWS-Konten, siehe Bereitstellung des Zugangs zu AWS-Konten im Besitz von Dritten in der IAM-Benutzerhandbuch.
- 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.

Überprüfung der Konformität für AWS Dienst für parallele Datenverarbeitung

Unser neues AWS Das Anmeldeerlebnis ist nicht für regulierte Arbeitslasten konzipiert. Wenn Sie unser neues verwenden AWS Erfahrung mit der Registrierung, die Sie aber nutzen möchten AWS für regulierte Arbeitslasten können Sie melden Sie sich an für AWS (fortgeschritten) oder aktiviere erweiterte Funktionen für dein AWS Umwelt.

Um zu erfahren, ob ein AWS-Service fällt in den Geltungsbereich bestimmter Compliance-Programme, siehe AWS-Services im Geltungsbereich des Compliance-Programms und wählen Sie das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme.

Sie können Prüfberichte von Drittanbietern mithilfe von AWS Artifact. Weitere Informationen finden Sie unter Berichte werden heruntergeladen in AWS Artifact.

Ihre Compliance-Verantwortung bei der Verwendung AWS-Services hängt von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens sowie den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Compliance-Verantwortung bei der Verwendung AWS-Services, siehe AWS Dokumentation zur Sicherheit.

Widerstandsfähigkeit in AWS Dienst für parallele Datenverarbeitung

Die AWS Die globale Infrastruktur basiert auf AWS-Regionen und Verfügbarkeitszonen. AWS-Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Für weitere Informationen über AWS-Regionen und Availability Zones finden Sie unter AWS Weltweite Infrastruktur.

Sicherheit der Infrastruktur in AWS Dienst für parallele Datenverarbeitung

Als verwalteter Dienst AWS Parallel Computing Service ist geschützt durch AWS globale Netzwerksicherheit. Für Informationen über AWS Sicherheitsdienste und wie AWS schützt die Infrastruktur, siehe AWS Cloud-Sicherheit. Um Ihre zu entwerfen AWS Umgebung, in der die bewährten Methoden für die Infrastruktursicherheit verwendet werden, finden Sie unter Schutz der Infrastruktur in Säule „Sicherheit“ AWS Gut durchdachtes Framework.

Du benutzt AWS veröffentlichte API-Aufrufe für den Zugriff AWS PCS über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungssammlungen 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.

Wann AWS PCS erstellt einen Cluster, der Dienst startet den Slurm-Controller in einem diensteigenen Konto, getrennt von den Rechenknoten in Ihrem Konto. Um die Kommunikation zwischen dem Controller und den Rechenknoten zu überbrücken, AWS PCS erstellt ein kontoübergreifendes Elastic Network Interface (ENI) in Ihrer VPC. Der Slurm-Controller verwendet das ENI, um die Rechenknoten auf verschiedenen Ebenen zu verwalten und mit ihnen zu kommunizieren AWS-Konten, um die Sicherheit und Isolierung der Ressourcen zu gewährleisten und gleichzeitig ein effizientes HPC zu ermöglichen und AI/ML Operationen.

Analyse und Management von Sicherheitslücken in AWS Dienst für parallele Datenverarbeitung

Konfiguration und IT-Steuerung liegen in der gemeinsamen Verantwortung von AWS und du. Weitere Informationen finden Sie auf der AWS Modell der geteilten Verantwortung. AWS erledigt grundlegende Sicherheitsaufgaben für die dem Dienstkonto zugrunde liegende Infrastruktur, z. B. das Patchen des Betriebssystems auf Controller-Instanzen, die Firewallkonfiguration und AWS Notfallwiederherstellung der Infrastruktur. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Einzelheiten finden Sie unter Bewährte Methoden für Sicherheit, Identität und Compliance.

> **Note**
>
> Slurm-Controller sind nicht verfügbar, solange wir sie aktualisieren. Laufende Jobs sind nicht betroffen. Jobs, die gesendet werden, wenn der Controller des Clusters nicht verfügbar ist, werden zurückgehalten, bis der Controller verfügbar ist.

Sie sind verantwortlich für die Sicherheit der zugrunde liegenden Infrastruktur in Ihrem AWS-Konto:

- Pflegen Sie Ihren Code, einschließlich Updates und Sicherheitspatches.
- Patchen und aktualisieren Sie das Betriebssystem im Amazon Machine Image (AMI) für Ihre Rechenknotengruppen und aktualisieren Sie Ihre Rechenknotengruppen, um das aktualisierte AMI zu verwenden.
- Aktualisieren Sie den Scheduler, um die unterstützten Versionen beizubehalten. Aktualisieren Sie das AMI für Ihre Rechenknotengruppen und aktualisieren Sie Ihre Rechenknotengruppe, um das aktualisierte AMI zu verwenden.
- Authentifizieren und verschlüsseln Sie die Kommunikation zwischen Benutzer-Clients und den Knoten, mit denen sie eine Verbindung herstellen.

Weitere Informationen zum Aktualisieren des AMI für Ihre Compute-Knotengruppen finden Sie unter Amazon Machine Images (AMIs) für AWS STK..

Cross-service verwirrte stellvertretende 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 AWS, dienstübergreifender Identitätswechsel kann zu einem verwirrten Stellvertreterproblem führen. Cross-service Identitätswechsel kann auftreten, wenn ein Dienst (der Dienst anrufen) ruft einen anderen Dienst an (den gerufener 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 Verwendung des `aws:SourceArn` und `aws:SourceAccount` globale Bedingungskontextschlüssel in Ressourcenrichtlinien, um die Berechtigungen einzuschränken,

die AWS Dienst für parallele Datenverarbeitung (AWS PCS) stellt der Ressource einen weiteren Dienst zur Verfügung. 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 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`*`:*`.

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.

Der Wert von `aws:SourceArn` muss ein Cluster-ARN sein.

Das folgende Beispiel zeigt, wie Sie den verwenden können `aws:SourceArn` und `aws:SourceAccount` globale Bedingungskontextschlüssel in AWS PCS, um das verwirrte Stellvertreterproblem zu verhindern.

```
{
"Version": "2012-10-17",
  "Statement": {
"Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "pcs.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": [
          "arn:aws:pcs:us-east-1:123456789012:cluster/*"
        ]
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
```

```
    }
  }
}
```

IAM-Rolle für Amazon EC2-Instances, die als Teil einer Rechenknotengruppe bereitgestellt werden

AWS PCS orchestriert automatisch die Amazon EC2-Kapazität für jede der konfigurierten Rechenknotengruppen in einem Cluster. Beim Erstellen einer Compute-Knotengruppe müssen Benutzer ein IAM-Instanzprofil über das bereitstellen `iamInstanceProfileArn` Feld. Das Instanzprofil gibt die Berechtigungen an, die den bereitgestellten EC2-Instances zugeordnet sind. AWS PCS akzeptiert jede Rolle, die `AWSPCS` als Rollennamenpräfix oder `/aws-pcs/` als Teil des Rollenpfads. Der `iam:PassRole` Die Berechtigung ist für die IAM-Identität (Benutzer oder Rolle) erforderlich, die eine Compute-Knotengruppe erstellt oder aktualisiert. Wenn ein Benutzer den aufruft `CreateComputeNodeGroup` oder `UpdateComputeNodeGroup` API-Aktionen, AWS PCS prüft, ob der Benutzer das ausführen darf `iam:PassRole` Aktion.

Die folgende Beispielrichtlinie gewährt die Berechtigung, nur IAM-Rollen weiterzugeben, deren Name mit `AWSPCS` beginnt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/AWSPCS*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Bewährte Sicherheitsmethoden für AWS Dienst für parallele Datenverarbeitung

In diesem Abschnitt werden bewährte Sicherheitsmethoden beschrieben, die sich speziell auf folgende Bereiche beziehen AWS Dienst für parallele Datenverarbeitung (AWS STÜCK). Weitere Informationen zu bewährten Sicherheitsmethoden finden Sie unter AWS, siehe Bewährte Methoden für Sicherheit, Identität und Compliance.

AMI-related Sicherheit

- Benutze nicht AWS PCS-Beispiel-AMIs für Produktionsworkloads. Die Beispiel-AMIs werden nicht unterstützt und sind nur für Testzwecke vorgesehen.
- Aktualisieren Sie regelmäßig das Betriebssystem und die Software im AMI für Ihre Rechenknotengruppen, um Sicherheitslücken zu minimieren.
- Verwenden Sie nur authentifizierte offizielle Dokumente AWS PCS-Pakete, die von official heruntergeladen wurden AWS Quellen.
- Regelmäßig aktualisieren AWS PCS-Pakete im AMI für Rechenknotengruppen und Aktualisierung der Rechenknoten, um das aktualisierte AMI zu verwenden. Erwägen Sie, diesen Prozess zu automatisieren, um Sicherheitslücken zu minimieren.

Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Sicherheit im Slurm Workload Manager

- Implementieren Sie Zugriffskontrollen und Netzwerkbeschränkungen, um die Steuerungs- und Rechenknoten von Slurm zu sichern. Erlauben Sie nur vertrauenswürdigen Benutzern und Systemen, Jobs einzureichen und auf Slurm-Verwaltungsbefehle zuzugreifen.
- Verwenden Sie die integrierten Sicherheitsfunktionen von Slurm, wie z. B. die Slurm-Authentifizierung, um sicherzustellen, dass Auftragseingaben und Kommunikation authentifiziert werden.
- Aktualisieren Sie die Slurm-Versionen, um einen reibungslosen Betrieb und Cluster-Unterstützung aufrechtzuerhalten.

> **Important**
>
> Jeder Cluster, der eine Version von Slurm verwendet, die erreicht wurde Ende der Lebensdauer des Supports (EOSL) wird sofort gestoppt. Verwenden Sie den Link oben auf den Seiten mit dem Benutzerhandbuch, um das zu abonnieren AWS RSS-Feed für die PCS-Dokumentation, um Benachrichtigungen zu erhalten, wenn sich eine Slurm-Version EOSL nähert.

Weitere Informationen finden Sie unter Slurm-Versionen in AWS 5 STCK.

- Rotation der Clustergeheimnisse erfolgt regelmäßig, um die Einhaltung der Sicherheitsvorschriften zu gewährleisten und potenzielle Sicherheitslücken zu beheben. Dies ist für die HIPAA- und FedRAMP-Konformität erforderlich.

  Weitere Informationen finden Sie unter Rotierende Clustergeheimnisse in AWS PCS.

Überwachung und Protokollierung

- Verwenden Sie Amazon CloudWatch Protokolle und AWS CloudTrail um Aktionen in Ihren Clustern zu überwachen und aufzuzeichnen und AWS-Konto. Verwenden Sie die Daten zur Problembehandlung und Prüfung.

Netzwerksicherheit

- Stellen Sie Ihr bereit AWS PCS-Cluster in einer separaten VPC, um Ihre HPC-Umgebung vom anderen Netzwerkverkehr zu isolieren.
- Verwenden Sie Sicherheitsgruppen und Netzwerkzugriffskontrolllisten (ACLs), um den eingehenden und ausgehenden Datenverkehr zu kontrollieren AWS PCS-Instanzen und Subnetze.
- Verwenden AWS PrivateLink oder VPC-Endpunkte, um den Netzwerkverkehr zwischen Ihren Clustern und anderen zu übertragen AWS Dienste innerhalb der AWS Netzwerk. Weitere Informationen finden Sie unter Zugriff AWS Parallel Computing Service mithilfe eines Schnittstellenendpunkts (AWS PrivateLink).

Protokollierung und Überwachung für AWS 5 STCK

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS PCS und Ihren anderen AWS-Ressourcen. AWS bietet die folgenden Überwachungstools, um AWS PCS zu überwachen, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Messwerte Ihrer Amazon EC2-Instances CloudWatch verfolgen lassen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- Amazon CloudWatch Logs ermöglicht es Ihnen, Ihre Protokolldateien von Amazon EC2-Instances und anderen Quellen aus zu überwachen CloudTrail, zu speichern und darauf zuzugreifen. CloudWatch Logs können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.
- AWS CloudTrailerfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt werden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3-Bucket. Sie können identifizieren, welche Benutzer und Konten aufgerufen wurden AWS, die Quell-IP-Adresse, von der aus die Aufrufe getätigt wurden, und wann die Aufrufe erfolgten. Weitere Informationen finden Sie im AWS CloudTrail -Benutzerhandbuch.

Die Auftragsabwicklung meldet sich an AWS 5 STCK

In den Protokollen zur Auftragsabwicklung erhalten Sie wichtige Informationen zu Ihren AWS PCS-Aufträgen (AWS Parallel Computing Service), sobald diese abgeschlossen sind, ohne dass zusätzliche Kosten anfallen. Sie können andere AWS Dienste verwenden, um auf Ihre Protokolldaten zuzugreifen und sie zu verarbeiten, z. B. Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3) und Amazon Data Firehose. AWS PCS zeichnet Metadaten zu Ihren Aufträgen auf, wie z. B. die folgenden.

- Job-ID und Name

- Benutzer- und Gruppeninformationen
- Status der Tätigkeit (z. B.`COMPLETED`,`FAILED`,`CANCELLED`)
- Verwendete Partition
- Zeitlimits
- Start-, Ende-, Absende- und berechtigte Zeiten
- Liste und Anzahl der Knoten
- Anzahl der Prozessoren
- Arbeitsverzeichnis
- Ressourcenverbrauch (CPU, Speicher)
- Ausgangscodes
- Knotendetails (Namen, Instanz-IDs, Instanztypen)

Inhalt

- Voraussetzungen
- Richten Sie Protokolle für die Fertigstellung von Aufträgen ein
- Wie finde ich Protokolle zur Fertigstellung von Aufträgen
  - CloudWatch Protokolle
  - Amazon S3
- Protokollfelder für den Abschluss eines Auftrags
- Beispiele für Protokolle zur Fertigstellung von Aufträgen

Voraussetzungen

Der IAM-Principal, der den AWS PCS-Cluster verwaltet, muss die `pcs:AllowVendedLogDeliveryForResource` Aktion zulassen.

Das folgende Beispiel für eine IAM-Richtlinie gewährt die erforderlichen Berechtigungen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
```

```
      {
         "Sid": "PcsAllowVendedLogsDelivery",
         "Effect": "Allow",
         "Action": ["pcs:AllowVendedLogDeliveryForResource"],
         "Resource": [
            "arn:aws:pcs:*::cluster/*"
         ]
      }
   ]
}
```

Richten Sie Protokolle für die Fertigstellung von Aufträgen ein

Mit dem AWS-Managementkonsole oder können Sie Protokolle zum Abschluss von Aufträgen für Ihren AWS PCS-Cluster einrichten AWS CLI.

AWS-Managementkonsole

So richten Sie Protokolle für den Abschluss von Aufträgen mit der Konsole ein

1. Öffnen Sie die AWS PCS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, dem Sie Protokolle zur Auftragsabwicklung hinzufügen möchten.
4. Wählen Sie auf der Cluster-Detailseite die Registerkarte Logs aus.
5. Wählen Sie unter Job Completion Logs die Option Hinzufügen aus, um bis zu 3 Log-Lieferziele aus CloudWatch Logs, Amazon S3 und Firehose hinzuzufügen.
6. Wählen Sie Protokolllieferungen aktualisieren aus.

AWS CLI

Um Protokolle für die Fertigstellung von Aufträgen einzurichten, verwenden Sie AWS CLI

1. Erstellen Sie ein Ziel für die Protokollzustellung:

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

Ersetzen Sie:

- *region*— Der AWS-Region Ort, an dem Sie das Ziel erstellen möchten, z. B. `us-east-1`
- *pcs-logs-destination*— Ein Name für das Ziel
- *resource-arn*— Der Amazon-Ressourcenname (ARN) einer CloudWatch Logs-Protokollgruppe, eines S3-Buckets oder eines Firehose-Lieferstreams.

Weitere Informationen finden Sie PutDeliveryDestination in der Amazon CloudWatch Logs API-Referenz.

2. Stellen Sie den PCS-Cluster als Quelle für die Protokollübermittlung ein:

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_JOBCOMP_LOGS
```

Ersetzen Sie:

- *region*— Die AWS-Region Ihres Clusters, wie `us-east-1`
- *cluster-logs-source-name*— Ein Name für die Quelle
- *cluster-arn*— der ARN Ihres AWS PCS-Clusters

Weitere Informationen finden Sie PutDeliverySource in der Amazon CloudWatch Logs API-Referenz.

3. Verbinden Sie die Lieferquelle mit dem Lieferziel:

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

Ersetzen Sie:

- *region*— Die AWS-Region, wie `us-east-1`
- *cluster-logs-source*— Der Name Ihrer Lieferquelle
- *destination-arn*— Die ARN Ihres Lieferziels

Weitere Informationen finden Sie CreateDelivery in der Amazon CloudWatch Logs API-Referenz.

Wie finde ich Protokolle zur Fertigstellung von Aufträgen

Sie können Protokollziele in CloudWatch Logs und Amazon S3 konfigurieren. AWS PCS verwendet die folgenden strukturierten Pfadnamen und Dateinamen.

CloudWatch Protokolle

AWS PCS verwendet das folgende Namensformat für den CloudWatch Logs-Stream:

```
AWSLogs/PCS/cluster-id/jobcomp.log
```

Beispiel: `AWSLogs/PCS/pcs_abc123de45/jobcomp.log`

Amazon S3

AWS PCS verwendet das folgende Namensformat für den S3-Pfad:

```
AWSLogs/account-id/PCS/region/cluster-id/jobcomp/year/month/day/hour/
```

Beispiel: `AWSLogs/111122223333/PCS/us-east-1/pcs_abc123de45/jobcomp/2025/06/19/11/`

AWS PCS verwendet das folgende Namensformat für die Protokolldateien:

```
PCS_jobcomp_year-month-day-hour_cluster-id_random-id.log.gz
```

Beispiel: `PCS_jobcomp_2025-06-19-11_pcs_abc123de45_04be080b.log.gz`

Protokollfelder für den Abschluss eines Auftrags

AWS PCS schreibt Protokolldaten zur Auftragsabwicklung als JSON-Objekte. Jeder Protokolleintrag enthält Metadatenfelder der obersten Ebene und ein `fields` Objekt. Die Felder der obersten Ebene identifizieren den Cluster, die Ereigniszeit und die Scheduler-Version. Das `fields` Objekt enthält die Auftragsdetails. Einige Felder im `fields` Objekt werden nur für Array-Jobs oder heterogene Jobs angezeigt.

In der folgenden Tabelle werden die Felder der obersten Ebene in jedem Protokolleintrag beschrieben.

Top-level Felder

Name	Beispielwert	Erforderlich	Hinweise
`resource_id`	`"pcs_22l8nzr3t9"`	Ja	Die AWS PCS-Cluster-ID
`resource_type`	`"PCS_CLUSTER"`	Ja	Immer `"PCS_CLUSTER"`
`event_timestamp`	`1750370337`	Ja	Unix-Epoche in Sekunden, in denen das Ereignis eingetreten ist
`scheduler_type`	`"slurm"`	Ja	Der Scheduler-Typ für den Cluster
`scheduler_major_version`	`"25.11"`	Ja	Die Hauptversion des Schedulers

In der folgenden Tabelle werden die Felder innerhalb des `fields` Objekts beschrieben.

Felder innerhalb des Feldobjekts

Name	Beispielwert	Erforderlich	Hinweise
`job_id`	`11`	Ja	Immer mit Wert vorhanden
`user`	`"root"`	Ja	Immer mit Wert präsent
`user_id`	`0`	Ja	Immer mit Wert präsent
`group`	`"root"`	Ja	Immer mit Wert präsent
`group_id`	`0`	Ja	Immer mit Wert präsent
`name`	`"wrap"`	Ja	Immer mit Wert präsent

Name	Beispielwert	Erforderlich	Hinweise
job_state	"COMPLETED"	Ja	Immer mit Wert präsent
partition	"MpiQueue-abcdef01-7"	Ja	Immer mit Wert präsent
time_limit	"UNLIMITED"	Ja	Immer präsent, könnte es aber sein "UNLIMITED"
start_time	"2025-06-19T10:58:57"	Ja	Immer präsent, könnte es aber sein "Unknown"
end_time	"2025-06-19T10:58:57"	Ja	Immer präsent, könnte es aber sein "Unknown"
node_list	"MpiNG-abcdef01-2345-1"	Ja	Immer präsent und wertvoll
node_cnt	1	Ja	Immer mit Wert präsent
proc_cnt	1	Ja	Immer mit Wert präsent
work_dir	"/root"	Ja	Immer präsent, könnte es aber sein "Unknown"
reservation_name	"weekly_maintenance"	Ja	Immer vorhanden, könnte aber eine leere Zeichenfolge sein ""
tres.cpu	1	Ja	Immer mit Wert vorhanden
tres.mem.val	600	Ja	Immer mit Wert präsent
tres.mem.unit	"M"	Ja	Kann sein "M" oder "bb"

Name	Beispielwert	Erforderlich	Hinweise
`tres.node`	`1`	Ja	Immer mit Wert präsent
`tres.billing`	`1`	Ja	Immer mit Wert präsent
`account`	`"finance"`	Ja	Immer vorhanden, kann aber eine leere Zeichenfolge sein `""`
`qos`	`"normal"`	Ja	Immer vorhanden, kann aber eine leere Zeichenfolge sein `""`
`wc_key`	`"project_1"`	Ja	Immer vorhanden, kann aber eine leere Zeichenfolge sein `""`
`cluster`	`"unknown"`	Ja	Immer präsent, könnte aber sein `"unknown"`
`submit_time`	`"2025-06-19T10:55:46"`	Ja	Immer präsent, könnte es aber sein `"Unknown"`
`eligible_time`	`"2025-06-19T10:55:46"`	Ja	Immer präsent, könnte es aber sein `"Unknown"`
`array_job_id`	`12`	Nein	Nur vorhanden, wenn es sich bei dem Job um einen Array-Job handelt
`array_task_id`	`1`	Nein	Nur vorhanden, wenn es sich bei dem Job um einen Array-Job handelt
`het_job_id`	`10`	Nein	Nur vorhanden, wenn es sich bei dem Job um einen heterogenen Job handelt

Name	Beispielwert	Erforderlich	Hinweise
`het_job_offset`	`0`	Nein	Nur vorhanden, wenn es sich bei der Stelle um eine heterogene Stelle handelt
`derived_exit_code_status`	`0`	Ja	Immer mit Mehrwert präsent
`derived_exit_code_signal`	`0`	Ja	Immer mit Wert präsent
`exit_code_status`	`0`	Ja	Immer mit Wert präsent
`exit_code_signal`	`0`	Ja	Immer mit Wert präsent
`node_details[0].name`	`"MpiNG-abcdef01-2345-1"`	Nein	Immer präsent, `node_details` könnte es aber sein `"[]"`
`node_details[0].instance_id`	`"i-0abcdef01234567a"`	Nein	Immer präsent, `node_details` könnte es aber sein `"[]"`
`node_details[0].instance_type`	`"t4g.micro"`	Nein	Immer präsent, `node_details` könnte es aber sein `"[]"`

Beispiele für Protokolle zur Fertigstellung von Aufträgen

Die folgenden Beispiele zeigen Jobabschlussprotokolle für verschiedene Auftragstypen und Status:

```
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750350777, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 1, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name":
 "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7", "time_limit":
 "UNLIMITED", "start_time": "2025-06-19T16:32:57", "end_time": "2025-06-19T16:33:03",
 "node_list": "MpiNG-abcdef01-2345-[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir":
 "/usr/bin", "reservation_name": "", "tres": { "cpu": 2, "mem": { "val": 1944,
 "unit": "M" }, "node": 2, "billing": 2 }, "account": "", "qos": "", "wc_key":
 "", "cluster": "unknown", "submit_time": "2025-06-19T16:29:40", "eligible_time":
 "2025-06-19T16:29:41", "derived_exit_code_status": 0, "derived_exit_code_signal":
 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name":
 "MpiNG-abcdef01-2345-1", "instance_id": "i-0abc123def45678", "instance_type":
 "t4g.micro" }, { "name": "MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901",
 "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750350793, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 2, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name":
 "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7", "time_limit":
 "UNLIMITED", "start_time": "2025-06-19T16:33:13", "end_time": "2025-06-19T16:33:14",
 "node_list": "MpiNG-abcdef01-2345-[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir":
 "/usr/bin", "reservation_name": "", "tres": { "cpu": 2, "mem": { "val": 1944,
 "unit": "M" }, "node": 2, "billing": 2 }, "account": "", "qos": "", "wc_key":
 "", "cluster": "unknown", "submit_time": "2025-06-19T16:33:13", "eligible_time":
 "2025-06-19T16:33:13", "derived_exit_code_status": 0, "derived_exit_code_signal":
 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name":
 "MpiNG-abcdef01-2345-1", "instance_id": "i-0abc123def45678", "instance_type":
 "t4g.micro" }, { "name": "MpiNG-abcdef01-2345-2", "instance_id": "i-0def456abc78901",
 "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750373937, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 3, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name":
 "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7", "time_limit":
 "UNLIMITED", "start_time": "2025-06-19T22:58:57", "end_time": "2025-06-19T22:58:57",
 "node_list": "MpiNG-abcdef01-2345-1", "node_cnt": 1, "proc_cnt": 1, "work_dir":
 "/root", "reservation_name": "", "tres": { "cpu": 1, "mem": { "val": 972,
 "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos": "", "wc_key":
 "", "cluster": "unknown", "submit_time": "2025-06-19T22:55:46", "eligible_time":
 "2025-06-19T22:55:46", "derived_exit_code_status": 0, "derived_exit_code_signal":
 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name":
 "MpiNG-abcdef01-2345-1", "instance_id": "i-0abc234def56789", "instance_type":
 "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374267, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
```

```
 { "job_id": 4, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name":
 "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7", "time_limit":
 "525600", "start_time": "2025-06-19T23:04:27", "end_time": "2025-06-19T23:04:27",
 "node_list": "MpiNG-abcdef01-2345-[1-2]", "node_cnt": 2, "proc_cnt": 2, "work_dir":
 "/root", "reservation_name": "", "tres": { "cpu": 2, "mem": { "val": 1944,
 "unit": "M" }, "node": 2, "billing": 2 }, "account": "", "qos": "", "wc_key":
 "", "cluster": "unknown", "submit_time": "2025-06-19T23:01:38", "eligible_time":
 "2025-06-19T23:01:38", "derived_exit_code_status": 0, "derived_exit_code_signal":
 0, "exit_code_status": 0, "exit_code_signal": 0, "node_details": [ { "name":
 "MpiNG-abcdef01-2345-1", "instance_id": "i-0abc234def56789", "instance_type":
 "t4g.micro" }, { "name": "MpiNG-abcdef01-2345-2", "instance_id": "i-0def345abc67890",
 "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374540, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 5, "user": "root", "user_id": 0, "group": "root", "group_id": 0, "name":
 "wrap", "job_state": "FAILED", "partition": "MpiQueue-abcdef01-7", "time_limit":
 "UNLIMITED", "start_time": "2025-06-19T23:09:00", "end_time": "2025-06-19T23:09:00",
 "node_list": "(null)", "node_cnt": 0, "proc_cnt": 0, "work_dir": "/root",
 "reservation_name": "", "tres": { "cpu": 1, "mem": { "val": 1, "unit": "G" }, "node":
 1, "billing": 1 }, "account": "", "qos": "", "wc_key": "", "cluster": "unknown",
 "submit_time": "2025-06-19T23:09:00", "eligible_time": "2025-06-19T23:09:00",
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374576, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 6, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "CANCELLED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:09:36", "end_time":
 "2025-06-19T23:09:36", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
 { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
 "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:09:35",
 "eligible_time": "2025-06-19T23:09:36", "het_job_id": 6, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374603, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 7, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "CANCELLED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:10:03", "end_time":
 "2025-06-19T23:10:03", "node_list": "(null)", "node_cnt": 0, "proc_cnt":
 0, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1, "mem":
 { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "", "qos":
 "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:10:03",
```

```
 "eligible_time": "2025-06-19T23:10:03", "het_job_id": 7, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status": 0,
 "exit_code_signal": 1, "node_details": [] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374684, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 8, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
 "2025-06-19T23:11:24", "node_list": "MpiNG-abcdef01-2345-1", "node_cnt": 1,
 "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1,
 "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
 "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374684, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 9, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:11:24", "end_time":
 "2025-06-19T23:11:24", "node_list": "MpiNG-abcdef01-2345-2", "node_cnt": 1,
 "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1,
 "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:11:23",
 "eligible_time": "2025-06-19T23:11:23", "het_job_id": 8, "het_job_offset": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "MpiNG-abcdef01-2345-2",
 "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374744, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 10, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
 "2025-06-19T23:12:24", "node_list": "MpiNG-abcdef01-2345-1", "node_cnt": 1,
 "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1,
 "mem": { "val": 400, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
 "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 0,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc234def56789", "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750374744, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
```

```
 { "job_id": 11, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:12:24", "end_time":
 "2025-06-19T23:12:24", "node_list": "MpiNG-abcdef01-2345-2", "node_cnt": 1,
 "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1,
 "mem": { "val": 600, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:12:14",
 "eligible_time": "2025-06-19T23:12:14", "het_job_id": 10, "het_job_offset": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "MpiNG-abcdef01-2345-2",
 "instance_id": "i-0def345abc67890", "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750376877, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 13, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:57", "end_time":
 "2025-06-19T23:47:58", "node_list": "MpiNG-abcdef01-2345-1", "node_cnt": 1,
 "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1,
 "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
 "eligible_time": "2025-06-19T23:43:56", "array_job_id": 12, "array_task_id": 1,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
{ "resource_id": "pcs_abc123de45", "resource_type": "PCS_CLUSTER", "event_timestamp":
 1750376878, "scheduler_type": "slurm", "scheduler_major_version": "25.11", "fields":
 { "job_id": 12, "user": "root", "user_id": 0, "group": "root", "group_id": 0,
 "name": "wrap", "job_state": "COMPLETED", "partition": "MpiQueue-abcdef01-7",
 "time_limit": "UNLIMITED", "start_time": "2025-06-19T23:47:58", "end_time":
 "2025-06-19T23:47:58", "node_list": "MpiNG-abcdef01-2345-1", "node_cnt": 1,
 "proc_cnt": 1, "work_dir": "/root", "reservation_name": "", "tres": { "cpu": 1,
 "mem": { "val": 972, "unit": "M" }, "node": 1, "billing": 1 }, "account": "",
 "qos": "", "wc_key": "", "cluster": "unknown", "submit_time": "2025-06-19T23:43:56",
 "eligible_time": "2025-06-19T23:43:56", "array_job_id": 12, "array_task_id": 2,
 "derived_exit_code_status": 0, "derived_exit_code_signal": 0, "exit_code_status":
 0, "exit_code_signal": 0, "node_details": [ { "name": "MpiNG-abcdef01-2345-1",
 "instance_id": "i-0abc345def67890", "instance_type": "t4g.micro" } ] } }
```

Scheduler loggt sich in AWS PCS ein

Sie können AWS PCS so konfigurieren, dass detaillierte Protokolldaten von Ihrem Cluster-Scheduler an Amazon CloudWatch Logs, Amazon Simple Storage Service (Amazon S3) und Amazon Data Firehose gesendet werden. Dies kann bei der Überwachung und Fehlerbehebung hilfreich sein.

AWS PCS übermittelt Logs von den folgenden Slurm-Daemons über den `PCS_SCHEDULER_LOGS` Protokolltyp:

- **`slurmctld`**— Der Slurm-Controller-Daemon. Verfügbar für alle unterstützten Slurm-Versionen.
- **`slurmdbd`**— Der Slurm-Datenbank-Daemon. Verfügbar für Slurm 24.11 und höher.
- **`slurmrestd`**— Der Slurm-REST-API-Daemon. Verfügbar für Slurm 25.05 und höher.

Cluster, für die der `PCS_SCHEDULER_LOGS` Versand bereits konfiguriert ist, beginnen automatisch mit dem Empfang `slurmdbd` und `slurmrestd` protokollieren, wenn sie eine unterstützte Slurm-Version ausführen. Es ist keine zusätzliche Konfiguration erforderlich.

Inhalt

- Voraussetzungen
- Richten Sie Scheduler-Protokolle ein
- Pfade und Namen von Log-Streams im Scheduler
- Beispiel für Scheduler-Protokolldatensätze

Voraussetzungen

Der IAM-Principal, der den AWS PCS-Cluster verwaltet, muss die `pcs:AllowVendedLogDeliveryForResource` Aktion zulassen.

Die folgende Beispiel-IAM-Richtlinie gewährt die erforderlichen Berechtigungen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "PcsAllowVendedLogsDelivery",
```

```
            "Effect": "Allow",
            "Action": ["pcs:AllowVendedLogDeliveryForResource"],
            "Resource": [
               "arn:aws:pcs:*::cluster/*"
            ]
         }
      ]
}
```

Richten Sie Scheduler-Protokolle ein

Sie können Scheduler-Protokolle für Ihren AWS PCS-Cluster mit dem AWS-Managementkonsole oder einrichten. AWS CLI

AWS-Managementkonsole

Um Scheduler-Logs mit der Konsole einzurichten

1. Öffnen Sie die AWS PCS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, zu dem Sie Scheduler-Logs hinzufügen möchten.
4. Wählen Sie auf der Seite mit den Cluster-Details die Registerkarte Logs aus.
5. Wählen Sie unter Scheduler Logs die Option Hinzufügen aus, um bis zu 3 Log-Lieferziele aus CloudWatch Logs, Amazon S3 und Firehose hinzuzufügen.
6. Wählen Sie Protokollzustellungen aktualisieren aus.

AWS CLI

Um Scheduler-Logs einzurichten mit AWS CLI

1. Erstellen Sie ein Ziel für die Protokollzustellung:

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

Ersetzen Sie:

- *region*— Der AWS-Region Ort, an dem Sie das Ziel erstellen möchten, z. B. `us-east-1`
- *pcs-logs-destination*— Ein Name für das Ziel
- *resource-arn*— Der Amazon-Ressourcenname (ARN) einer CloudWatch Logs-Protokollgruppe, eines S3-Buckets oder eines Firehose-Lieferstreams.

Weitere Informationen finden Sie PutDeliveryDestinationin der Amazon CloudWatch Logs API-Referenz.

2. Legen Sie den PCS-Cluster als Quelle für die Protokollzustellung fest:

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_SCHEDULER_LOGS
```

Ersetzen Sie:

- *region*— Die AWS-Region Ihres Clusters, wie `us-east-1`
- *cluster-logs-source-name*— Ein Name für die Quelle
- *cluster-arn*— der ARN Ihres AWS PCS-Clusters

Weitere Informationen finden Sie PutDeliverySourcein der Amazon CloudWatch Logs API-Referenz.

3. Connect die Lieferquelle mit dem Lieferziel:

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

Ersetzen Sie:

- *region*— Die AWS-Region, wie `us-east-1`
- *cluster-logs-source*— Der Name Ihrer Lieferquelle
- *destination-arn*— Die ARN Ihres Lieferziels

Weitere Informationen finden Sie CreateDeliveryin der Amazon CloudWatch Logs API-Referenz.

Pfade und Namen von Log-Streams im Scheduler

Der Pfad und der Name der AWS PCS-Scheduler-Protokolle hängen vom Zieltyp ab.

Der `${log_name}` Wert in den folgenden Pfaden ist`slurmctld`, oder `slurmdbd``slurmrestd`, abhängig vom Daemon, der das Protokoll erstellt hat.

- CloudWatch Protokolle
  - Ein CloudWatch Logs-Stream folgt dieser Namenskonvention.

    ```
    AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}.log
    ```

    Example

    ```
    AWSLogs/PCS/abcdef0123/slurmctld_25.11.log
    AWSLogs/PCS/abcdef0123/slurmdbd_24.11.log
    AWSLogs/PCS/abcdef0123/slurmrestd_25.05.log
    ```
- S3 bucket
  - Ein S3-Bucket-Ausgabepfad folgt dieser Namenskonvention:

    ```
    AWSLogs/${account-id}/PCS/${region}/${cluster_id}/${log_name}/
    ${scheduler_major_version}/yyyy/MM/dd/HH/
    ```

    Example

    ```
    AWSLogs/111111111111/PCS/us-east-2/abcdef0123/slurmctld/25.11/2024/09/01/00/
    AWSLogs/111111111111/PCS/us-east-2/abcdef0123/slurmdbd/24.11/2024/09/01/00/
    AWSLogs/111111111111/PCS/us-east-2/abcdef0123/slurmrestd/25.05/2024/09/01/00/
    ```
  - Ein S3-Objektname folgt dieser Konvention:

    ```
    PCS_${log_name}_${scheduler_major_version}_#{expr date 'event_timestamp', format:
     "yyyy-MM-dd-HH"}_${cluster_id}_${hash}.log
    ```

Example

```
PCS_slurmctld_25.11_2024-09-01-00_abcdef0123_0123abcdef.log
```

Beispiel für Scheduler-Protokolldatensätze

AWS Die PCS-Scheduler-Protokolle sind strukturiert. Sie enthalten Felder wie die Cluster-ID, den Scheduler-Typ, Haupt- und Patch-Versionen sowie die Protokollnachricht, die vom Slurm-Daemon-Prozess ausgegeben wird. Die `node_type` Felder `log_name` und identifizieren, welcher Daemon das Protokoll erstellt hat.

Das folgende Beispiel zeigt einen `slurmctld` Protokolleintrag.

```
{
    "resource_id": "s3431v9rx2",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1721230979,
    "log_level": "info",
    "log_name": "slurmctld",
    "scheduler_type": "slurm",
    "scheduler_major_version": "25.11",
    "scheduler_patch_version": "2",
    "node_type": "controller_primary",
    "message": "[2024-07-17T15:42:58.614+00:00] Running as primary controller\n"
}
```

Das folgende Beispiel zeigt einen `slurmdbd` Protokolldatensatz (Slurm 24.11 und höher).

```
{
    "resource_id": "pcs_bu93qsds2j",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1774485082772,
    "log_level": "info",
    "log_name": "slurmdbd",
    "scheduler_type": "slurm",
    "scheduler_major_version": "25.11",
    "scheduler_patch_version": "2",
    "node_type": "slurmdbd_primary",
    "message": "[2026-03-26T00:31:22.772+00:00] mysql_common: storage token refreshed"
```

```
}
```

Das folgende Beispiel zeigt einen `slurmrestd` Protokolldatensatz (Slurm 25.05 und höher).

```
{
    "resource_id": "pcs_bu93qsds2j",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1774485082772,
    "log_level": "info",
    "log_name": "slurmrestd",
    "scheduler_type": "slurm",
    "scheduler_major_version": "25.05",
    "scheduler_patch_version": "3",
    "node_type": "slurmrestd_primary",
    "message": "[2026-03-26T00:31:22.772+00:00] slurmrestd: Listening on port 6820\n"
}
```

Scheduler-Auditprotokolle in AWS PCS

In den Auditprotokollen des Schedulers werden RPC-Operationen (Remote Procedure Call) aufgezeichnet, die vom Slurm-Controller () und dem Datenbank-Daemon (`slurmctld`) Ihres Clusters verarbeitet werden. `slurmdbd` Das `AUDIT_RPCS:` Präfix in der Protokollnachricht identifiziert diese Protokolle. Sie unterstützen Anwendungsfälle zur Sicherheitsüberprüfung und zur Einhaltung von Vorschriften.

Für Cluster, auf denen Slurm 25.11 und höher ausgeführt wird, stellt AWS PCS Audit-Logs getrennt nach `PCS_SCHEDULER_AUDIT_LOGS` Protokolltyp bereit. Durch diese Trennung können Sie die Aufnahme- und Speicherkosten für Audit-Logs unabhängig von Ihren Betriebsprotokollen kontrollieren, da Audit-Logs bis zu 90% des Scheduler-Protokollvolumens ausmachen können.

> **Note**
>
> Bei Clustern, auf denen Slurm-Versionen vor 25.11 ausgeführt werden, bleiben die Audit-Logs erhalten `PCS_SCHEDULER_LOGS` und der `PCS_SCHEDULER_AUDIT_LOGS` Protokolltyp ist nicht verfügbar. Weitere Informationen zu Scheduler-Logs finden Sie unter. Scheduler loggt sich in AWS PCS ein

Inhalt

- Voraussetzungen
- Richten Sie Scheduler-Auditprotokolle ein
- Die Pfade und Namen von Log-Streams werden von Scheduler geprüft
- Beispiel für einen Scheduler-Audit-Logeintrag
- Verhalten des Audit-Logs nach Slurm-Version

Voraussetzungen

Bevor Sie Scheduler-Audit-Logs einrichten können, müssen Sie die folgenden Anforderungen erfüllen:

- Auf Ihrem Cluster muss Slurm 25.11 oder höher ausgeführt werden.
- Der IAM-Prinzipal, der den AWS PCS-Cluster verwaltet, muss die Aktion zulassen. `pcs:AllowVendedLogDeliveryForResource`

Die folgende Beispiel-IAM-Richtlinie gewährt die erforderlichen Berechtigungen.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "PcsAllowVendedLogsDelivery",
         "Effect": "Allow",
         "Action": ["pcs:AllowVendedLogDeliveryForResource"],
         "Resource": [
            "arn:aws:pcs:*::cluster/*"
         ]
      }
   ]
}
```

Richten Sie Scheduler-Auditprotokolle ein

Sie können Scheduler-Auditprotokolle für Ihren AWS PCS-Cluster mit dem AWS-Managementkonsole oder einrichten. AWS CLI Scheduler-Prüfprotokolle sind optional. AWS PCS liefert sie erst, wenn Sie sie abonnieren.

AWS-Managementkonsole

Um Scheduler-Audit-Logs mit der Konsole einzurichten

1. Öffnen Sie die AWS PCS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, dem Sie Scheduler-Audit-Logs hinzufügen möchten.
4. Wählen Sie auf der Seite mit den Cluster-Details die Registerkarte Logs aus.
5. Wählen Sie unter Scheduler Audit Logs die Option Hinzufügen aus, um bis zu 3 Log-Lieferziele aus CloudWatch Logs, Amazon S3 und Firehose hinzuzufügen.
6. Wählen Sie Protokollzustellungen aktualisieren aus.

AWS CLI

Um Scheduler-Audit-Logs einzurichten mit AWS CLI

1. Erstellen Sie ein Ziel für die Protokollzustellung:

```
aws logs put-delivery-destination --region region \
  --name pcs-audit-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

Ersetzen Sie:

- *region*— Der AWS-Region Ort, an dem Sie das Ziel erstellen möchten, z. B. `us-east-1`
- *pcs-audit-logs-destination*— Ein Name für das Ziel
- *resource-arn*— Der Amazon-Ressourcenname (ARN) einer CloudWatch Logs-Protokollgruppe, eines S3-Buckets oder eines Firehose-Lieferstreams.

Weitere Informationen finden Sie PutDeliveryDestinationin der Amazon CloudWatch Logs API-Referenz.

2. Legen Sie den PCS-Cluster als Quelle für die Protokollzustellung fest:

```
aws logs put-delivery-source --region region \
  --name cluster-audit-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_SCHEDULER_AUDIT_LOGS
```

Ersetzen Sie:

- *region*— Die AWS-Region Ihres Clusters, wie `us-east-1`
- *cluster-audit-logs-source-name*— Ein Name für die Quelle
- *cluster-arn*— der ARN Ihres AWS PCS-Clusters

Weitere Informationen finden Sie PutDeliverySourcein der Amazon CloudWatch Logs API-Referenz.

3. Connect die Lieferquelle mit dem Lieferziel:

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-audit-logs-source \
  --delivery-destination-arn destination-arn
```

Ersetzen Sie:

- *region*— Die AWS-Region, wie `us-east-1`
- *cluster-audit-logs-source*— Der Name Ihrer Lieferquelle
- *destination-arn*— Die ARN Ihres Lieferziels

Weitere Informationen finden Sie CreateDeliveryin der Amazon CloudWatch Logs API-Referenz.

Die Pfade und Namen von Log-Streams werden von Scheduler geprüft

Der Pfad und der Name der AWS PCS-Scheduler-Audit-Logs hängen vom Zieltyp ab.

- CloudWatch Protokolle
  - Ein CloudWatch Logs-Stream folgt dieser Namenskonvention.

    ```
    AWSLogs/PCS/${cluster_id}/${log_name}_${scheduler_major_version}_audit.log
    ```

    Wo `${log_name}` ist `slurmctld` oder`slurmdbd`.

    Example

    ```
    AWSLogs/PCS/abcdef0123/slurmctld_25.11_audit.log
    AWSLogs/PCS/abcdef0123/slurmdbd_25.11_audit.log
    ```

- S3 bucket
  - Ein S3-Bucket-Ausgabepfad folgt dieser Namenskonvention:

    ```
    AWSLogs/${account-id}/PCS/${region}/${cluster_id}/scheduler_audit/${log_name}/yyyy/
    MM/dd/HH/
    ```

    Example

    ```
    AWSLogs/111111111111/PCS/us-east-2/abcdef0123/scheduler_audit/
    slurmctld/2026/03/01/00/
    AWSLogs/111111111111/PCS/us-east-2/abcdef0123/scheduler_audit/
    slurmdbd/2026/03/01/00/
    ```

Beispiel für einen Scheduler-Audit-Logeintrag

AWS Die Auditprotokolle von PCS Scheduler sind strukturiert. Sie verwenden dasselbe Schema wie Scheduler-Logs, wobei die Log-Meldung das Präfix enthält. `AUDIT_RPCS:` Hier ist ein Beispiel von`slurmctld`.


```
{
    "resource_id": "pcs_bu93qsds2j",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1774481175953,
    "log_level": "info",
    "log_name": "slurmctld",
    "scheduler_type": "slurm",
    "scheduler_major_version": "25.11",
```

```
    "scheduler_patch_version": "2",
    "node_type": "controller_primary",
    "message": "[2026-01-21T08:19:26.692+00:00] AUDIT_RPCS: [slurmctld-
primary:6817(fd:18)] msg_type=REQUEST_PARTITION_INFO uid=0 client=[10.0.76.95:56918]\n"
}
```

Hier ist ein Beispiel von`slurmdbd`.

```
{
    "resource_id": "pcs_bu93qsds2j",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1774485082772,
    "log_level": "info",
    "log_name": "slurmdbd",
    "scheduler_type": "slurm",
    "scheduler_major_version": "25.11",
    "scheduler_patch_version": "2",
    "node_type": "slurmdbd_primary",
    "message": "[2026-01-21T08:19:26.692+00:00] AUDIT_RPCS: msg_type=DBD_GET_CLUSTERS
 uid=0 client=[28.5.0.18:36658] protocol=11008\n"
}
```

Verhalten des Audit-Logs nach Slurm-Version

In der folgenden Tabelle wird beschrieben, wie Audit-Logs je nach der Slurm-Version, die auf Ihrem Cluster ausgeführt wird, bereitgestellt werden.

Slurm-Version	**PCS_SCHEDULER_LOGS** enthält	**PCS_SCHEDULER_AUDIT_LOGS** verfügbar
Vor 25.11	Alle Protokolle, einschließlich Audit-Logs	Nein
25.11 und später	Nur Betriebsprotokolle (Auditprotokolle wurden entfernt)	Ja (Opt-In)

Scaling meldet sich an AWS STCK

Durch Skalierungsprotokolle können Sie aufzeichnen, wie AWS Parallel Computing Service skaliert die Rechenknotengruppen in Ihrem Cluster. Jeder Protokolleintrag zeichnet einen Zustandsübergang für einen Rechenknoten auf: zum Beispiel einen Instance-Start, eine Knotenregistrierung, eine Herunterskalierung oder einen Startfehler mit der entsprechenden Ursache. Sie können Skalierungsprotokolle verwenden, um Fragen zu beantworten, z. B. warum eine Knotengruppe ihre Zielgröße nicht erreicht hat, welche Starts aus Kapazitätsgründen fehlgeschlagen sind und wann ein bestimmter Knoten gestartet oder gestoppt wurde.

Sie können Skalierungsprotokolle an Amazon senden CloudWatch, Amazon Simple Storage Service (Amazon S3) und Amazon Data Firehose. Für jeden Übergang AWS PCS zeichnet die folgenden Daten auf:

- Gruppen-Identifikatoren für Cluster und Rechenknoten
- Der Knotenname
- Der neue Status
- Ein Ursachencode und eine für Menschen lesbare Beschreibung
- Die EC2-Instanz-ID, wenn eine Instance mit dem Übergang verknüpft ist
- Starten Sie gegebenenfalls den Kontext, z. B. den Instance-Typ und das Subnetz

AWS PCS liefert Skalierungsprotokolle über das `PCS_SCALING_LOGS` Protokolltyp. Die Skalierung der Protokollzustellung ist optional. AWS PCS liefert keine Skalierungsprotokolle für einen Cluster, bis Sie eine Bereitstellung dafür konfiguriert haben.

Inhalt

- Voraussetzungen
- Richten Sie Skalierungsprotokolle ein
- Skalierungsprotokolle finden
  - CloudWatch Protokolle
  - Amazon S3
- Skalieren von Protokollfeldern
- Statuscodes und Ursachencodes
- Beispiel für Skalierungsprotokolle

Voraussetzungen

Der IAM-Principal, der das verwaltet AWS Der PCS-Cluster muss Folgendes zulassen `pcs:AllowVendedLogDeliveryForResource` Aktion.

Die folgende Beispiel-IAM-Richtlinie gewährt die erforderlichen Berechtigungen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PcsAllowVendedLogsDelivery",
            "Effect": "Allow",
            "Action": ["pcs:AllowVendedLogDeliveryForResource"],
            "Resource": [
                "arn:aws:pcs:*::cluster/*"
            ]
        }
    ]
}
```

Richten Sie Skalierungsprotokolle ein

Sie können Skalierungsprotokolle für Ihre einrichten AWS PCS-Cluster mit dem AWS-Managementkonsole oder AWS CLI.

AWS-Managementkonsole

Um Skalierungsprotokolle mit der Konsole einzurichten

1. Öffnen Sie das AWS PCS-Konsole.
2. Klicken Sie im Navigationsbereich auf Cluster.
3. Wählen Sie den Cluster aus, dem Sie Skalierungsprotokolle hinzufügen möchten.
4. Wählen Sie auf der Seite mit den Cluster-Details die Protokolle Registerkarte.
5. Unter Protokolle skalieren, wählen Hinzufügen um bis zu 3 Log-Lieferziele aus einer Liste hinzuzufügen CloudWatch Logs, Amazon S3 und Firehose.
6. Wählen Sie Log-Lieferungen aktualisieren.

AWS CLI

Um Skalierungsprotokolle einzurichten mit dem AWS CLI

1. Erstellen Sie ein Ziel für die Protokollübermittlung:

```
aws logs put-delivery-destination --region region \
  --name pcs-logs-destination \
  --delivery-destination-configuration \
  destinationResourceArn=resource-arn
```

Ersetzen Sie:

- *region* — Das AWS-Region wo Sie das Ziel erstellen möchten, z. B. `us-east-1`
- *pcs-logs-destination* — Ein Name für das Ziel
- *resource-arn* — Der Amazon-Ressourcenname (ARN) eines CloudWatch Protokolliert die Protokollgruppe, den S3-Bucket oder den Firehose-Lieferstream.

Weitere Informationen finden Sie unter PutDeliveryDestination in der Amazonas CloudWatch API-Referenz für Protokolle.

2. Stellen Sie den PCS-Cluster als Quelle für die Protokollübermittlung ein:

```
aws logs put-delivery-source --region region \
  --name cluster-logs-source-name \
  --resource-arn cluster-arn \
  --log-type PCS_SCALING_LOGS
```

Ersetzen Sie:

- *region* — Das AWS-Region Ihres Clusters, wie `us-east-1`
- *cluster-logs-source-name* — Ein Name für die Quelle
- *cluster-arn* — der ARN Ihres AWS PCS-Cluster

Weitere Informationen finden Sie unter PutDeliverySource in der Amazonas CloudWatch API-Referenz für Protokolle.

3. Verbinden Sie die Lieferquelle mit dem Lieferziel:

```
aws logs create-delivery --region region \
  --delivery-source-name cluster-logs-source \
  --delivery-destination-arn destination-arn
```

Ersetzen Sie:

- *region* — Das AWS-Region, wie `us-east-1`
- *cluster-logs-source* — Der Name Ihrer Lieferquelle
- *destination-arn* — Die ARN Ihres Lieferziels

Weitere Informationen finden Sie unter CreateDelivery in der Amazonas CloudWatch API-Referenz für Protokolle.

Skalierungsprotokolle finden

Sie können Protokollziele konfigurieren in CloudWatch Protokolle und Amazon S3. AWS PCS verwendet die folgenden strukturierten Pfadnamen und Dateinamen.

CloudWatch Protokolle

AWS PCS verwendet das folgende Namensformat für CloudWatch Protokolliert den Stream:

```
AWSLogs/PCS/cluster-id/node_lifecycle.log
```

Beispiel: `AWSLogs/PCS/pcs_abc123de45/node_lifecycle.log`

Amazon S3

AWS PCS verwendet das folgende Namensformat für den S3-Pfad:

```
AWSLogs/account-id/PCS/region/cluster-id/node_lifecycle/year/month/day/hour/
```

Beispiel: `AWSLogs/111122223333/PCS/us-east-1/pcs_abc123de45/node_lifecycle/2026/09/03/11/`

AWS PCS verwendet das folgende Namensformat für die Protokolldateien:

```
PCS_cluster-id_node_lifecycle_year-month-day-hour_random-id.log.gz
```

Beispiel: `PCS_pcs_abc123de45_node_lifecycle_2026-09-03-11_04be080b.log.gz`

Skalieren von Protokollfeldern

AWS PCS schreibt skalierende Protokolldaten als JSON-Objekte. Jeder Protokolleintrag enthält Metadatenfelder auf oberster Ebene und eine `details` Objekt. Die Felder der obersten Ebene identifizieren den Cluster, die Ereigniszeit und die Compute-Knotengruppen- und Scheduler-Knoten-IDs. Die `details` Objekt enthält die Übergangsdetails.

In der folgenden Tabelle werden die Felder der obersten Ebene in jedem Protokolleintrag beschrieben.

Top-level Felder

Name	Beispielwert	Erforderlich	Hinweise
`resource_id`	`"pcs_22l8nzr3t9"`	Ja	Die AWS PCS-Cluster-ID
`resource_type`	`"PCS_CLUSTER"`	Ja	Immer `"PCS_CLUSTER"`
`event_timestamp`	`1789179816460`	Ja	Unix-Epoche in Millisekunden, als das Ereignis eintrat
`compute_node_group_id`	`"pcs_56zr33g8"`	Ja	Die Rechenknotengruppe, zu der der Knoten gehört
`scheduler_node_id`	`"compute-1"`	Ja	Die Scheduler-Knoten-ID
`status_code`	`"LAUNCH_FAILED"`	Ja	Der neue Status des Knotens. Siehe Statuscodes und Ursachencodes
`reason_code`	`"INSUFFICIENT_CAPACITY"`	Ja	Warum ist der Übergang passiert
`description`	`"The launch failed`	Ja	Eine für Menschen lesbare Zusammenfassung, die aus den

Name	Beispielwert	Erforderlich	Hinweise
	`because EC2 did not have enough capacity"`		Status- und Ursachencodes abgeleite t wurde
`details`	`{...}`	Nein	Zusätzlicher Kontext für den Übergang. Die Schlüssel hängen vom Übergangstyp ab

In der folgenden Tabelle werden die Schlüssel beschrieben, die in der `details` Objekt.

Felder innerhalb der **`Details`** Objekt

Name	Beispielwert	Erforderlich	Hinweise
`instance_id`	`"i-0abcde f01234567 a"`	Nein	Die dem Knoten zugeordnete EC2-Instanz. Fehlt, wenn keine Instanz zugeordnet ist, z. B. nach einem Startfehler
`ec2_error _code`	`"Insuffic ientInsta nceCapaci ty"`	Nein	Der EC2-Fehlercode für einen fehlgeschlagenen Start. Derselbe Code erscheint in der CloudTrail Aufzeichnung des CreateFleet rufen Sie Ihr Konto an
`instance_ type`	`"t4g.micro"`	Nein	Der gestartete Instanztyp
`subnet_id`	`"subnet-0 abc123"`	Nein	Das Subnetz, auf das der Start abzielt

Statuscodes und Ursachencodes

Das `status_code` Das Feld gibt den Status an, in den der Knoten eingegeben hat. Das `reason_code` Das Feld sagt dir warum. Zum Beispiel ein `LAUNCH_FAILED` Status mit einem

`INSUFFICIENT_CAPACITY` Grund bedeutet, dass Amazon EC2 im Zielsubnetz keine Kapazität für den angeforderten Instance-Typ bereitstellen konnte. A) `LIMIT_EXCEEDED` bedeutet, dass ein Amazon EC2-Kontingent den Start blockiert hat.

Statuscodes und Ursachencodes

Statuscode	Ursachencode	Description
`LAUNCHED`	`SCHEDULER_REQUESTED`	Die Instanz wurde gestartet, um eine Scale-Up-Anfrage des Schedulers zu erfüllen
`LAUNCHED`	`MAINTAIN_MIN_CAPACITY`	Die Instanz wurde gestartet, um die Mindestkapazität der Knotengruppe aufrechtzuerhalten
`REGISTERED`	`REGISTER_SUCCESS`	Der Knoten wurde erfolgreich über den PCS-Agenten registriert
`ACTIVE`	`BOOTSTRAP_SUCCESS`	Der Knoten-Bootstrap ist abgeschlossen; slurmd läuft und akzeptiert Jobs
`PENDING_REPLACEMENT`	`NODE_GROUP_UPDATE`	Der Knoten ist leer, bevor er für ein Knotengruppen-Update heruntergefahren wird
`PENDING_REPLACEMENT`	`EC2_HEALTH_CHECK_FAILED`	Der Knoten ist überlastet, weil eine Integritätsprüfung des Knotens oder seiner Instanz fehlgeschlagen ist
`PENDING_REPLACEMENT`	`EC2_SCHEDULED_EVENT`	Der Knoten ist überlastet, weil für seine Instanz ein geplantes EC2-Wartungsereignis stattfindet

Statuscode	Ursachencode	Description
LAUNCH_FAILED	INSUFFICIENT_CAPACITY	Der Knoten hat keine Instance erhalten, da nicht genügend Kapazität verfügbar war
LAUNCH_FAILED	LIMIT_EXCEEDED	Der Instanzstart ist fehlgeschlagen, weil ein Kontingent überschritten wurde
LAUNCH_FAILED	LAUNCH_ERROR	Der Knoten konnte seine Instanz nicht starten
TERMINATED	SCHEDULER_REQUESTED	Die Instanz wurde beendet, nachdem der Scheduler den Knoten angehalten hatte (Timeout im Leerlauf oder vom Benutzer veranlasstes Herunterfahren)
TERMINATED	UNHEALTHY_NODE	Die Instanz wurde beendet, weil der Scheduler den Knoten als fehlerhaft gemeldet hat
TERMINATED	BOOTSTRAP_FAILED	Die Instanz wurde beendet, weil der Knoten nicht gestartet werden konnte
TERMINATED	EXTERNALLY_TERMINATED	Die Instanz wurde außerhalb von PCS beendet. Der Knoten wurde freigegeben, sodass er ersetzt werden kann
TERMINATED	REPLACEMENT_REQUESTED	Die Instanz wurde beendet, weil der Knoten für den Austausch leer war

Statuscode	Ursachencode	Description
TERMINATED	NODE_GROUP_DELETION	Die Instanz wurde im Rahmen des Löschens der Compute-K notengruppe beendet
TERMINATE_FAILED	EC2_OPERATION_NOT_ PERMITTED	Das Beenden der Instanz ist fehlgeschlagen. Stellen Sie sicher, dass der Kündigung sschutz auf der Instance nicht aktiviert ist
TERMINATE_FAILED	EC2_ERROR	Die Beendigung der Instanz ist fehlgeschlagen

Beispiel für Skalierungsprotokolle

Das folgende Beispiel zeigt einen erfolgreichen Instance-Start:

```
{
    "resource_id": "pcs_abc123de45",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1789179816460,
    "compute_node_group_id": "pcs_abcdef01",
    "scheduler_node_id": "compute-1",
    "status_code": "LAUNCHED",
    "reason_code": "SCHEDULER_REQUESTED",
    "description": "Instance launched to satisfy a scheduler scale-up request.",
    "details": {
        "instance_id": "i-0abc12345def67890",
        "instance_type": "c6i.xlarge",
        "subnet_id": "subnet-0123456789abcdef0"
    }
}
```

Das folgende Beispiel zeigt einen Startfehler, der durch unzureichende EC2-Kapazität verursacht wurde. Nein `instance_id` ist vorhanden, weil keine Instanz gestartet wurde.

```
{
```

```
    "resource_id": "pcs_abc123de45",
    "resource_type": "PCS_CLUSTER",
    "event_timestamp": 1789179816460,
    "compute_node_group_id": "pcs_abcdef01",
    "scheduler_node_id": "compute-2",
    "status_code": "LAUNCH_FAILED",
    "reason_code": "INSUFFICIENT_CAPACITY",
    "description": "Node did not get an instance because there was not enough capacity
 available.",
    "details": {
        "ec2_error_code": "InsufficientInstanceCapacity"
    }
}
```

Überwachen AWS Paralleler Rechendienst mit Amazon CloudWatch

Amazon überwacht CloudWatch den Zustand und die Leistung Ihres AWS Parallel Computing Service (AWS PCS) -Clusters, indem in regelmäßigen Abständen Metriken aus dem Cluster erfasst werden. Diese Metriken werden beibehalten, sodass Sie auf historische Daten zugreifen und Einblicke in die Leistung Ihres Clusters im Laufe der Zeit gewinnen können.

CloudWatch ermöglicht es Ihnen auch, die von AWS PCS gestarteten EC2-Instances zu überwachen, um Ihre Skalierungsanforderungen zu erfüllen. Sie können zwar die Protokolle der laufenden Instances überprüfen, aber CloudWatch Metriken und Logging-Daten werden in der Regel gelöscht, sobald die Instances beendet werden. Sie können den CloudWatch Agenten auf Instances jedoch mithilfe einer EC2-Startvorlage so konfigurieren, dass Metriken und Protokolle auch nach dem Beenden der Instance beibehalten werden, sodass eine langfristige Überwachung und Analyse möglich ist.

Lesen Sie die Themen in diesem Abschnitt, um mehr über die Überwachung von AWS PCS zu erfahren. CloudWatch

Themen

- Überwachen AWS PCS-Metriken verwenden CloudWatch
- Überwachen AWS PCS-Instanzen, die Amazon verwenden CloudWatch

Überwachen AWS PCS-Metriken verwenden CloudWatch

Sie können den Zustand des AWS PCS-Clusters mithilfe von Amazon CloudWatch überwachen. Amazon sammelt Daten aus Ihrem Cluster und wandelt sie in Metriken nahezu in Echtzeit um. Diese Statistiken werden für einen Zeitraum von 15 Monaten aufbewahrt, sodass Sie auf historische Informationen zugreifen und sich einen besseren Überblick über die Leistung Ihres Clusters verschaffen können. Cluster-Metriken werden CloudWatch in Abständen von 1 Minute an gesendet. Weitere Informationen finden Sie CloudWatch unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch.

AWS PCS veröffentlicht die folgenden Metriken im AWS/PCS Namespace in CloudWatch. Sie haben eine einzige Dimension,`ClusterId`.

Name	Description	Einheiten
ActualCapacity	IdleCapacity + UtilizedCapacity	Anzahl
CapacityUtilization	UtilizedCapacity / ActualCapacity	Anzahl
DesiredCapacity	ActualCapacity + PendingCapacity	Anzahl
IdleCapacity	Anzahl der Instanzen, die zwar ausgeführt werden, aber keinen Aufträgen zugeordnet sind	Anzahl
UtilizedCapacity	Anzahl der Instanzen, die ausgeführt werden und Aufträgen zugewiesen sind	Anzahl

Überwachen AWS PCS-Instanzen, die Amazon verwenden CloudWatch

AWS PCS startet Amazon EC2-Instances nach Bedarf, um die in Ihren PCS-Rechenknotengruppen definierten Skalierungsanforderungen zu erfüllen. Sie können diese Instances mit Amazon CloudWatch überwachen, während sie ausgeführt werden. Sie können die Protokolle laufender

Instances überprüfen, indem Sie sich bei ihnen anmelden und interaktive Befehlszeilentools verwenden. Standardmäßig werden CloudWatch Metrikdaten jedoch nur für einen begrenzten Zeitraum aufbewahrt, sobald eine Instance beendet wird. Instance-Logs werden in der Regel zusammen mit den EBS-Volumes gelöscht, auf denen die Instance basiert. Um Metriken oder Logging-Daten der von PCS gestarteten Instances auch nach deren Beendigung beizubehalten, können Sie den CloudWatch Agenten auf Ihren Instances mit einer EC2-Startvorlage konfigurieren. Dieses Thema bietet einen Überblick über die Überwachung laufender Instances und enthält Beispiele für die Konfiguration persistenter Instance-Metriken und -Protokolle.

Überwachung laufender Instanzen

Suche nach AWS PCS-Instanzen

Um von PCS gestartete Instanzen zu überwachen, suchen Sie nach den laufenden Instances, die einem Cluster oder einer Rechenknotengruppe zugeordnet sind. Überprüfen Sie dann in der EC2-Konsole für eine bestimmte Instance die Abschnitte Status und Alarme sowie Monitoring. Wenn der Anmeldezugriff für diese Instances konfiguriert ist, können Sie eine Verbindung zu ihnen herstellen und verschiedene Protokolldateien auf den Instances überprüfen. Weitere Informationen zur Identifizierung der Instanzen, die von PCS verwaltet werden, finden Sie unterSuchen nach Rechenknotengruppeninstanzen in AWS PCS.

Aktivierung detaillierter Metriken

Standardmäßig werden Instance-Metriken in Intervallen von 5 Minuten erfasst. Um Metriken in Intervallen von einer Minute zu erfassen, aktivieren Sie die detaillierte CloudWatch Überwachung in Ihrer Vorlage für den Start der Compute-Node-Gruppe. Weitere Informationen finden Sie unter Aktivieren Sie die detaillierte CloudWatch Überwachung.

Konfiguration persistenter Instanz-Metriken und -Protokolle

Sie können die Metriken und Protokolle Ihrer Instances beibehalten, indem Sie den CloudWatch Amazon-Agenten auf ihnen installieren und konfigurieren. Dies besteht aus drei Hauptschritten:

1. Erstellen Sie eine CloudWatch Agentenkonfiguration.
2. Speichern Sie die Konfiguration dort, wo sie von PCS-Instanzen abgerufen werden kann.
3. Schreiben Sie eine EC2-Startvorlage, die die CloudWatch Agentensoftware installiert, Ihre Konfiguration abruft und den CloudWatch Agenten mithilfe der Konfiguration startet.

Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch unter Erfassung von Metriken, Protokollen und Ablaufverfolgungen mit dem CloudWatch Agenten und. Verwenden von Amazon EC2 EC2-Startvorlagen mit AWS PCS

Erstellen Sie eine CloudWatch Agentenkonfiguration

Bevor Sie den CloudWatch Agenten auf Ihren Instances bereitstellen, müssen Sie eine JSON-Konfigurationsdatei generieren, in der die zu erfassenden Metriken, Logs und Traces angegeben sind. Konfigurationsdateien können mit einem Assistenten oder manuell mit einem Texteditor erstellt werden. Die Konfigurationsdatei wird für diese Demonstration manuell erstellt.

Erstellen Sie auf einem Computer, auf dem die AWS-CLI installiert ist, eine CloudWatch Konfigurationsdatei mit dem Namen config.json mit den folgenden Inhalten. Sie können auch die folgende URL verwenden, um eine Kopie der Datei herunterzuladen.

```
https://aws-hpc-recipes.s3.amazonaws.com/main/recipes/pcs/cloudwatch/assets/config.json
```

Hinweise

- Die Protokollpfade in der Beispieldatei beziehen sich auf Amazon Linux 2023. Wenn Ihre Instances ein anderes Basisbetriebssystem verwenden, ändern Sie die Pfade entsprechend.
- Um andere Logs zu erfassen, fügen Sie zusätzliche Einträge unter hinzu`collect_list`.
- Die Werte in `{brackets}` sind als Vorlage dienende Variablen. Eine vollständige Liste der unterstützten Variablen finden Sie unter Manuelles Erstellen oder Bearbeiten der CloudWatch Agentenkonfigurationsdatei im CloudWatch Amazon-Benutzerhandbuch.
- Sie können wählen, `metrics` ob Sie diese Informationstypen weglassen `logs` oder nicht sammeln möchten.

```
{
    "agent": {
        "metrics_collection_interval": 60
    },
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/var/log/cloud-init.log",
                        "log_group_class": "STANDARD",
```

```
                        "log_group_name": "/PCSLogs/instances",
                        "log_stream_name": "{instance_id}.cloud-init.log",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/cloud-init-output.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.cloud-init-output.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/amazon/pcs/bootstrap.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.bootstrap.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/slurmd.log",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.slurmd.log",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/messages",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.messages",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    },
                    {
                        "file_path": "/var/log/secure",
                        "log_group_class": "STANDARD",
                        "log_stream_name": "{instance_id}.secure",
                        "log_group_name": "/PCSLogs/instances",
                        "retention_in_days": 30
                    }
                ]
            }
        }
    },
    "metrics": {
```

```
        "aggregation_dimensions": [
            [
                "InstanceId"
            ]
        ],
        "append_dimensions": {
            "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
            "ImageId": "${aws:ImageId}",
            "InstanceId": "${aws:InstanceId}",
            "InstanceType": "${aws:InstanceType}"
        },
        "metrics_collected": {
            "cpu": {
                "measurement": [
                    "cpu_usage_idle",
                    "cpu_usage_iowait",
                    "cpu_usage_user",
                    "cpu_usage_system"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ],
                "totalcpu": false
            },
            "disk": {
                "measurement": [
                    "used_percent",
                    "inodes_free"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
            "diskio": {
                "measurement": [
                    "io_time"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            },
```

```
            "mem": {
                "measurement": [
                    "mem_used_percent"
                ],
                "metrics_collection_interval": 60
            },
            "swap": {
                "measurement": [
                    "swap_used_percent"
                ],
                "metrics_collection_interval": 60
            }
        }
    }
}
```

Diese Datei weist den CloudWatch Agenten an, mehrere Dateien zu überwachen, die bei der Diagnose von Fehlern in den Bereichen Bootstrapping, Authentifizierung und Anmeldung sowie in anderen Bereichen zur Problembehandlung hilfreich sein können. Dazu zählen:

- `/var/log/cloud-init.log`— Ausgabe aus der Anfangsphase der Instanzkonfiguration
- `/var/log/cloud-init-output.log`— Ausgabe von Befehlen, die während der Instanzkonfiguration ausgeführt werden
- `/var/log/amazon/pcs/bootstrap.log`— Ausgabe von PCS-specific Vorgängen, die während der Instanzkonfiguration ausgeführt werden
- `/var/log/slurmd.log`— Ausgabe des Daemons slurmd des Slurm-Workload-Managers
- `/var/log/messages`— Systemnachrichten aus dem Kernel, den Systemdiensten und den Anwendungen
- `/var/log/secure`— Protokolle im Zusammenhang mit Authentifizierungsversuchen wie SSH, Sudo und anderen Sicherheitsereignissen

Die Protokolldateien werden an eine CloudWatch Protokollgruppe mit dem Namen `/PCSLogs/instances` gesendet. Die Log-Streams sind eine Kombination aus der Instanz-ID und dem Basisnamen der Logdatei. Die Protokollgruppe hat eine Aufbewahrungszeit von 30 Tagen.

Darüber hinaus weist die Datei den CloudWatch Agenten an, mehrere gängige Metriken zu sammeln und sie nach Instanz-ID zu aggregieren.

Speichern Sie die Konfiguration

Die CloudWatch Agentenkonfigurationsdatei muss an einem Ort gespeichert werden, an dem PCS-Rechenknoteninstanzen darauf zugreifen können. Dafür gibt es zwei gängige Methoden. Sie können es in einen Amazon S3-Bucket hochladen, auf den Ihre Compute-Knotengruppen-Instances über ihr Instance-Profil Zugriff haben. Alternativ können Sie ihn als SSM-Parameter im Amazon Systems Manager Parameter Store speichern.

Laden Sie es in einen S3-Bucket hoch

Verwenden Sie die folgenden AWS-CLI-Befehle, um Ihre Datei in S3 zu speichern. Bevor Sie den Befehl ausführen, nehmen Sie die folgenden Ersetzungen vor:

- Ersetzen Sie ihn *amzn-s3-demo-bucket* durch Ihren eigenen S3-Bucket-Namen

Erstellen Sie zunächst (dies ist optional, wenn Sie einen vorhandenen Bucket haben) einen Bucket für Ihre Konfigurationsdatei (en).

```
aws s3 mb s3://amzn-s3-demo-bucket
```

Laden Sie als Nächstes die Datei in den Bucket hoch.

```
aws s3 cp ./config.json s3://amzn-s3-demo-bucket/
```

Als SSM-Parameter speichern

Verwenden Sie den folgenden Befehl, um Ihre Datei als SSM-Parameter zu speichern. Bevor Sie den Befehl ausführen, nehmen Sie die folgenden Ersetzungen vor:

- *region-code*Ersetzen Sie es durch die AWS-Region, in der Sie mit AWS PCS arbeiten.
- (Optional) *AmazonCloudWatch-PCS* Ersetzen Sie den Parameter durch Ihren eigenen Namen. Beachten Sie, dass Sie, wenn Sie das Präfix des Namens von `AmazonCloudWatch-` ändern, dem SSM-Parameter in Ihrem Knotengruppen-Instanzprofil ausdrücklich Lesezugriff gewähren müssen.

```
aws ssm put-parameter \
   --region region-code \
   --name "AmazonCloudWatch-PCS" \
   --type String \
   --value file://config.json
```

Schreiben Sie eine EC2-Startvorlage

Die spezifischen Details für die Startvorlage hängen davon ab, ob Ihre Konfigurationsdatei in S3 oder SSM gespeichert ist.

Verwenden Sie eine in S3 gespeicherte Konfiguration

Dieses Skript installiert den CloudWatch Agenten, importiert eine Konfigurationsdatei aus einem S3-Bucket und startet den CloudWatch Agenten damit. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- *amzn-s3-demo-bucket*— Der Name eines S3-Buckets, aus dem Ihr Konto lesen kann
- */config.json*— Pfad relativ zum S3-Bucket-Stamm, in dem die Konfiguration gespeichert ist

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

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

packages:
- amazon-cloudwatch-agent

runcmd:
- aws s3 cp s3://amzn-s3-demo-bucket/config.json /etc/s3-cw-config.json
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
 ec2 -s -c file://etc/s3-cw-config.json

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

Das IAM-Instanzprofil für die Knotengruppe muss Zugriff auf den Bucket haben. Im obigen Benutzerdatenskript finden Sie ein Beispiel für eine IAM-Richtlinie für den Bucket.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

Beachten Sie auch, dass die Instances ausgehenden Datenverkehr zum S3 und CloudWatch zu den Endpunkten zulassen müssen. Dies kann je nach Clusterarchitektur mithilfe von Sicherheitsgruppen oder VPC-Endpunkten erreicht werden.

Verwenden Sie eine in SSM gespeicherte Konfiguration

Dieses Skript installiert den CloudWatch Agenten, importiert eine Konfigurationsdatei aus einem SSM-Parameter und startet den CloudWatch Agenten damit. Ersetzen Sie die folgenden Werte in diesem Skript durch Ihre eigenen Daten:

- (Optional) *AmazonCloudWatch-PCS* Ersetzen Sie den Parameter durch Ihren eigenen Namen.

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

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

packages:
- amazon-cloudwatch-agent

runcmd:
- /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m
 ec2 -s -c ssm:AmazonCloudWatch-PCS

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

Der IAM-Instanzrichtlinie für die Knotengruppe muss die CloudWatchAgentServerPolicy angehängt sein.

Wenn Ihr Parametername nicht mit beginnt, müssen `AmazonCloudWatch-` Sie dem SSM-Parameter in Ihrem Knotengruppen-Instanzprofil ausdrücklich Lesezugriff gewähren. Hier ist ein Beispiel für eine IAM-Richtlinie, die dies für das Präfix veranschaulicht. *DOC-EXAMPLE-PREFIX*

JSON

```
{
  "Version":"2012-10-17",
  "Statement" : [
    {
      "Sid" : "CustomCwSsmMParamReadOnly",
      "Effect" : "Allow",
      "Action" : [
        "ssm:GetParameter"
      ],
      "Resource" : "arn:aws:ssm:*:*:parameter/DOC-EXAMPLE-PREFIX*"
    }
  ]
}
```

Beachten Sie auch, dass die Instances ausgehenden Datenverkehr zum SSM und zu den Endpunkten zulassen müssen. CloudWatch Dies kann je nach Clusterarchitektur mithilfe von Sicherheitsgruppen oder VPC-Endpunkten erreicht werden.

Protokollieren von API-Aufrufen für den AWS Parallel Computing Service mit AWS CloudTrail

AWS PCS ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS PCS ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für AWS PCS als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS PCS-Konsole und Codeaufrufen für die AWS PCS-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS PCS. 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 CloudTrail, die an AWS PCS gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

AWS PCS-Informationen in CloudTrail

CloudTrail ist auf Ihrem aktiviert AWS-Konto , wenn Sie das Konto erstellen. Wenn in AWS PCS eine Aktivität auftritt, wird diese Aktivität zusammen mit anderen CloudTrail AWS Serviceereignissen im Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem anzeigen, suchen und herunterladen AWS-Konto. Weitere Informationen finden Sie unter Ereignisse mit dem CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem System AWS-Konto, einschließlich Ereignisse für AWS PCS, 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 anlegen, gilt dieser für alle AWS-Regionen-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 PCS-Aktionen werden von der AWS Parallel Computing Service API-Referenz protokolliert CloudTrail und sind in dieser dokumentiert. Beispielsweise generieren Aufrufe der `DeleteCluster` Aktionen `CreateComputeNodeGroupUpdateQueue`, und Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.

- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter CloudTrail -Element userIdentity.

Grundlegendes zu CloudTrail Protokolldateieinträgen von AWS PCS

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für eine `CreateQueue` Aktion.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
        "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
        "accountId": "012345678910",
        "accessKeyId": "ASIAY36PTPIEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAY36PTPIEEXAMPLE",
                "arn": "arn:aws:iam::012345678910:role/Admin",
                "accountId": "012345678910",
                "userName": "Admin"
            },
            "attributes": {
                "creationDate": "2024-07-16T17:05:51Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-07-16T17:13:09Z",
    "eventSource": "pcs.amazonaws.com",
```

```
    "eventName": "CreateQueue",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36",
    "requestParameters": {
        "clientToken": "c13b7baf-2894-42e8-acec-example",
        "clusterIdentifier": "abcdef0123",
        "computeNodeGroupConfigurations": [
            {
                "computeNodeGroupId": "abcdef0123"
            }
        ],
        "queueName": "all"
    },
    "responseElements": {
        "queue": {
            "arn": "arn:aws:pcs:us-east-1:609783872011:cluster/abcdef0123/queue/
abcdef0123",
            "clusterId": "abcdef0123",
            "computeNodeGroupConfigurations": [
                {
                    "computeNodeGroupId": "abcdef0123"
                }
            ],
            "createdAt": "2024-07-16T17:13:09.276069393Z",
            "id": "abcdef0123",
            "modifiedAt": "2024-07-16T17:13:09.276069393Z",
            "name": "all",
            "status": "CREATING"
        }
    },
    "requestID": "a9df46d7-3f6d-43a0-9e3f-example",
    "eventID": "7ab18f88-0040-47f5-8388-example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "012345678910",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "pcs.us-east-1.amazonaws.com"
    },
```

```
    "sessionCredentialFromConsole": "true"
}
```

Endpunkte und Servicekontingente für AWS 5 STÜCK

In den folgenden Abschnitten werden die Endpunkte und Servicekontingente für beschrieben AWS Dienst für parallele Datenverarbeitung (AWS STÜCK). Servicequoten, früher bezeichnet als Grenzen, sind die maximale Anzahl von Serviceressourcen oder Operationen für Ihre AWS-Konto.

Ihr AWS-Konto hat Standardkontingente für jeden AWS Dienst. Sofern nicht anders angegeben, ist jedes Kontingent Region-specific. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Weitere Informationen finden Sie unter AWS Service Quotas in der Allgemeinen AWS -Referenz.

Inhalt

- Service-Endpunkte
- Servicekontingente
  - Interne Kontingente
  - Relevante Kontingente für andere AWS service

Service-Endpunkte

Das `pcs.region.api.aws` Die Endpunkte sind Dual-Stack. Sie unterstützen sowohl IPv4 als auch IPv6. Das `pcs.region.amazonaws.com` Endpunkte unterstützen nur IPv4. Weitere Informationen zu Dual-Stack-Endpunkten finden Sie unter Dual-stack Endpunkte in der AWS Allgemeine Referenz.

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Ohio)	us-east-2	pcs.us-east-2.amazonaws.com pcs-fips.us-east-2.amazonaws.com pcs-fips.us-east-2.api.aws pcs.us-east-2.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Nord-Virginia)	us-east-1	pcs.us-east-1.amazonaws.com pcs-fips.us-east-1.amazonaws.com pcs-fips.us-east-1.api.aws pcs.us-east-1.api.aws	HTTPS
USA West (Oregon)	us-west-2	pcs.us-west-2.amazonaws.com pcs-fips.us-west-2.amazonaws.com pcs-fips.us-west-2.api.aws pcs.us-west-2.api.aws	HTTPS
Afrika (Kapstadt)	af-south-1	pcs.af-south-1.amazonaws.com pcs.af-south-1.api.aws	HTTPS
Asien-Pazifik (Mumbai)	ap-south-1	pcs.ap-south-1.amazonaws.com pcs.ap-south-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Asien-Pazifik (Hyderabad)	ap-south-2	pcs.ap-south-2.amazonaws.com pcs.ap-south-2.api.aws	HTTPS
Asien-Pazifik (Singapur)	ap-southeast-1	pcs.ap-southeast-1.amazonaws.com pcs.ap-southeast-1.api.aws	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	pcs.ap-southeast-2.amazonaws.com pcs.ap-southeast-2.api.aws	HTTPS
Asien-Pazifik (Jakarta)	ap-southeast-3	pcs.ap-southeast-3.amazonaws.com pcs.ap-southeast-3.api.aws	HTTPS
Asien-Pazifik (Melbourne)	ap-southeast-4	pcs.ap-southeast-4.amazonaws.com pcs.ap-southeast-4.api.aws	HTTPS
Asien-Pazifik (Malaysia)	ap-southeast-5	pcs.ap-southeast-5.amazonaws.com pcs.ap-southeast-5.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Asien-Pazifik (Neuseeland)	ap-southeast-6	pcs.ap-southeast-6.amazonaws.com pcs.ap-southeast-6.api.aws	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	pcs.ap-northeast-1.amazonaws.com pcs.ap-northeast-1.api.aws	HTTPS
Asien-Pazifik (Seoul)	ap-northeast-2	pcs.ap-northeast-2.amazonaws.com pcs.ap-northeast-2.api.aws	HTTPS
Asia Pacific (Osaka)	ap-northeast-3	pcs.ap-northeast-3.amazonaws.com pcs.ap-northeast-3.api.aws	HTTPS
Kanada (Zentral)	ca-central-1	pcs.ca-central-1.amazonaws.com pcs.ca-central-1.api.aws	HTTPS
Europa (Frankfurt)	eu-central-1	pcs.eu-central-1.amazonaws.com pcs.eu-central-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Europa (Irland)	eu-west-1	pcs.eu-west-1.amazonaws.com pcs.eu-west-1.api.aws	HTTPS
Europa (London)	eu-west-2	pcs.eu-west-2.amazonaws.com pcs.eu-west-2.api.aws	HTTPS
Europa (Paris)	eu-west-3	pcs.eu-west-3.amazonaws.com pcs-eu-west-3.api.aws	HTTPS
Europa (Milan)	eu-south-1	pcs.eu-south-1.amazonaws.com pcs-eu-south-1.api.aws	HTTPS
Europa (Spain)	eu-south-2	pcs.eu-south-2.amazonaws.com pcs-eu-south-2.api.aws	HTTPS
Europa (Stockholm)	eu-north-1	pcs.eu-north-1.amazonaws.com pcs.eu-north-1.api.aws	HTTPS
Südamerika (São Paulo)	sa-east-1	pcs.sa-east-1.amazonaws.com pcs.sa-east-1.api.aws	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
AWS GovCloud (US-East)	us-gov-east-1	pcs.us-gov-east-1.amazonaws.com pcs-fips.us-gov-east-1.amazonaws.com pcs-fips.us-gov-east-1.api.aws pcs.us-gov-east-1.api.aws	HTTPS
AWS GovCloud (US-West)	us-gov-west-1	pcs.us-gov-west-1.amazonaws.com pcs-fips.us-gov-west-1.amazonaws.com pcs-fips.us-gov-west-1.api.aws pcs.us-gov-west-1.api.aws	HTTPS

Servicekontingente

Name	Standard	Einstellbar	Beschreibung
Cluster	5	Ja	Die maximale Anzahl von Clustern pro AWS-Region.

> **Note**
>
> Die Standardwerte sind die anfänglichen Kontingente, die von festgelegt wurden AWS. Diese Standardwerte sind unabhängig von den tatsächlich angewendeten Kontingentwerten und den maximal möglichen Servicekontingenten. Weitere Informationen finden Sie unter Terminologie in Service Quotas im Service Quotas User Guide.

Diese Servicekontingente sind aufgeführt unter AWS Paralleler Rechendienst (PCS) in der AWS-Managementkonsole. Informationen zum Beantragen einer Kontingenterhöhung für Werte, die als anpassbar angezeigt werden, finden Sie unter Eine Kontingenterhöhung beantragen in der Service Quotas — Benutzerhandbuch.

> **Important**
>
> Denken Sie daran, den Strom zu überprüfen AWS-Region Einstellung in der AWS-Managementkonsole.

Interne Kontingente

Die folgenden Kontingente sind interne Kontingente. Um eine Kontingenterhöhung zu beantragen, erstellen Sie einen Support-Fall in der AWS Support Center-Konsole.

Name	Standard	Einstellbar	Beschreibung
Gleichzeitige Cluster-Erstellung	1	Nein	Die maximale Anzahl von Clustern in der `Creating` Bundesstaat pro AWS-Region.
Knotengruppen pro Cluster berechnen	10	Ja (per Support-Fall)	Die maximale Anzahl von Rechenknotengruppen pro Cluster.

Name	Standard	Einstellbar	Beschreibung
Warteschlangen pro Cluster	10	Ja (per Support-Fall)	Die maximale Anzahl von Warteschlangen pro Cluster.

Relevante Kontingente für andere AWS service

AWS PCS verwendet andere AWS Dienstleistungen. Ihre Servicekontingente für diese Dienste wirken sich auf Ihre Nutzung von aus AWS STCK.

Amazon EC2-Servicequoten, die sich auswirken AWS STCK

- Spot-Instance-Anfragen
- Ausführen von Instances auf Abruf
- Startvorlagen
- Startvorlagenversionen
- Amazon EC2-API-Anfragen

Weitere Informationen finden Sie unter Amazon EC2-Servicekontingente in der Amazon Elastic Compute Cloud-Benutzerhandbuch.

Behebung von Problemen in AWS Dienst für parallele Datenverarbeitung

Die folgenden Themen bieten Anleitungen zur Behebung einiger Probleme, auf die Sie möglicherweise stoßen AWS STCK.

- Cluster-Aktualisierungen
- Aktualisierungen der Cluster-Version
- Bootstrap-Probleme mit Rechenknoten
- Benutzerdefinierte Slurm-Einstellungen
- EC2-Instanzen wurden nach dem Neustart beendet
- Identität und Zugriff
- Ungültige Knotenregistrierung (INVALID_REG)
- Einreichung des Jobs MaxJobCount Begrenzung
- Fehlende Buchhaltungsunterlagen
- Probleme beim Neustart von Slurm

Eine EC2-Instanz in AWS PCS wird nach dem Neustart beendet und ersetzt

Überblick über das Problem

Nach dem Neustart einer EC2-Instance in einer Compute-Knotengruppe AWS PCS beendet und ersetzt die Instance automatisch.

Warum passiert das

AWS PCS unterstützt keine Instanzneustarts. Wenn eine EC2-Instanz neu gestartet wird, AWS PCS betrachtet die Instance als fehlerhaft und ersetzt sie. Wenn AWS PCS beendet und ersetzt Ihre Instances kontinuierlich. Dies kann daran liegen, dass Ihre Instances nach dem Start neu gestartet werden. Einige Beispiele hierfür sind automatische Neustarts auf der EC2-Instance (z. B. ein automatischer Neustart nach dem Patchen), Automatisierung außerhalb der EC2-Instance (z. B. eine Netzwerkverwaltungsanwendung) und ein anderes AWS Dienst (wie AWS Systems Manager) oder ein manueller Neustart durch eine Person.

Vorgehensweise

Sie können Ihre überprüfen `slurmctld` oder `slurmd` protokolliert, um zu sehen, ob Ihre Instanz neu gestartet wurde. Weitere Informationen erhalten Sie unter Scheduler loggt sich in AWS PCS ein und Überwachen AWS PCS-Instanzen, die Amazon verwenden CloudWatch. Das folgende Beispiel `slurmctld` Ein Protokolleintrag gibt an, dass die Instanz neu gestartet wurde:

Example

```
[2024-09-12T06:42:50.393+00:00] validate_node_specs: Node Login-1 unexpectedly rebooted
 boot_time=1726123354 last response=1726123285
```

Neustart aufgrund von Patches

Nach der Installation von Patches ist häufig ein Neustart erforderlich. Wenden Sie Patches nicht direkt auf eine EC2-Instance an, die Teil von AWS PCS-Rechenknotengruppe. Wenn Sie Ihre EC2-Instances patchen müssen, sollten Sie Ihre Patches auf ein aktualisiertes Amazon Machine Image (AMI) anwenden und Ihre Rechenknotengruppen so aktualisieren, dass sie das aktualisierte AMI verwenden. Neue EC2-Instances, die AWS PCS-Starts für diese Rechenknotengruppen verwenden das aktualisierte (gepatchte) AMI. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Beheben Sie Probleme mit dem Bootstrap und der Registrierung von Rechenknoten in AWS STCK

Wenn Rechenknoten beim Bootstrap oder bei der Registrierung von Rechenknoten nicht ordnungsgemäß bei Ihrem AWS Bei einem PCS-Cluster können die folgenden Symptome auftreten:

- Jobs fangen nicht an
- Sie können keine Verbindung zu Instanzen in herstellen AWS Systems Manager
- Instanzen werden unerwartet heruntergefahren
- Instanzen werden kontinuierlich ersetzt

Diese Fehler können durch Probleme beim Start der EC2-Instance oder während des AWS PCS-Rechenknoten-Bootstrap-Prozess. In diesem Thema werden Verfahren beschrieben, die Ihnen bei der Behebung von Problemen während des AWS Bootstrap-Prozess für PCS-Knoten. Weitere Informationen zur Problembehandlung beim Start der EC2-Instance finden Sie unter Beheben

Sie Probleme beim Starten der Amazon EC2-Instance in der Amazon Elastic Compute Cloud-Benutzerhandbuch.

Bootstrap-Fehler treten auf, wenn eine EC2-Instance erfolgreich gestartet wird, aber während des Beitritts zum AWS PCS-Cluster. Der Bootstrap-Prozess umfasst zwei Hauptphasen:

- Registrierung des Knotens — Die EC2-Instance ruft die RegisterComputeNodeGroupInstance AWS PCS-API-Aktion zur Registrierung bei der AWS PCS-Dienst. Fehler können aufgrund von Problemen in den folgenden Bereichen auftreten:
  - Berechtigungen
    - Falsches Instanzprofil
  - Netzwerk
    - Es kann keine Verbindung hergestellt werden AWS PCS-Endpunkte
    - Falsch konfiguriert AWS PCS-Endpunkt
    - Instanz in einem öffentlichen Subnetz ohne öffentliche IP
    - Multi-NIC Instanz in einem öffentlichen Subnetz
  - Geheimer Cluster-Schlüssel
    - Der geheime Clusterschlüssel wurde gelöscht oder zum Löschen markiert
- Slurm-Integration — Die Instanz läuft `slurmd` und tritt dem Slurm-Cluster bei. Fehler können aufgrund von Problemen in den folgenden Bereichen auftreten:
  - Berechtigungen
    - Sicherheitsgruppenkonfiguration
    - Slurmctld kann den Rechenknoten nicht pingen
  - Benutzerdefiniertes AMI-Setup
    - Fehlende NVIDIA-Treiber
    - ResumeTimeout erreicht

Wie funktioniert Slurm auf AWS 2 STCK

Es könnte dir helfen, die Standardmethode von Slurm mit der Funktionsweise von Slurm zu vergleichen AWS 2 STCK.

Standardverarbeitung von Slurm-Aufträgen

Die folgenden Schritte finden bei der Standardverarbeitung von Slurm-Aufträgen statt:

1. Wenn Sie einen Job einreichen, `slurmctld` validiert den Job und stellt ihn in die Warteschlange.
2. Wenn Ressourcen verfügbar werden, `slurmctld` weist vorhandene Knoten zu.
3. `slurmd` Daemons führen Jobs auf zugewiesenen Knoten aus.

Die Verarbeitung von Slurm-Jobs ist aktiviert AWS 5 STCK

Die folgenden Schritte finden statt in AWS PCS-Auftragsverarbeitung:

1. Wenn Sie einen Job einreichen, `slurmctld` validiert den Job und stellt ihn in die Warteschlange.
2. Wenn zusätzliche Kapazität benötigt wird, AWS PCS verwendet die Startvorlage für die Compute-Knotengruppe, um neue EC2-Instances zu starten.
3. Neue Instanzen werden per Bootstrapping in den Cluster geladen:
   a. Instanzen registrieren sich bei AWS STCK.
   b. Instanzen treten dem Slurm-Cluster bei.
4. Wenn die Ressourcen bereit sind, `slurmctld` weist Knoten zu (einschließlich neu gestarteter Knoten).
5. `slurmd` Daemons führen Jobs auf zugewiesenen Knoten aus.

Rufen Sie Instanzprotokolle ab

Der erste Schritt bei der Behebung von Rechenknoten-Bootstrap-Problemen besteht darin, die Instanzprotokolle abzurufen. Sie können eine der folgenden Methoden verwenden:

AWS CLI

Rufen Sie die Konsolenausgabe mit dem folgenden Befehl vom Rechenknoten ab:

```
aws ec2 get-console-output --region us-east-1 --instance-id i-1234567890abcdef0 --
output text
```

Ersetzen *us-east-1* mit deinem AWS Region und *i-1234567890abcdef0* mit Ihrer Instanz-ID.

AWS Systems Manager

Wenn Sie mithilfe von Systems Manager eine Verbindung zur Instanz herstellen können, können Sie die Bootstrap-Protokolldatei direkt anzeigen:

1. Stellen Sie mithilfe von Systems Manager eine Verbindung zur Instanz her. Weitere Informationen finden Sie unter Eine Sitzung starten in der Systems Manager-Benutzerhandbuch.
2. Sehen Sie sich die Bootstrap-Protokolldatei an:

```
sudo cat /var/log/amazon/pcs/bootstrap.log
```

> **Note**
>
> Wenn während der Initialisierungsphase ein Problem auftritt, müssen Sie möglicherweise etwa 20 Minuten warten, bevor Sie eine Verbindung zur Instance herstellen können. Systems Manager- und SSH-Dienste werden erst gestartet, nachdem die Initialisierung abgeschlossen ist oder wenn bei der Bootstrap-Ausführung im Falle eines Fehlers ein Timeout erreicht wird.

Abrufen VPC/Subnet/Security Gruppen aus einer Instanz-ID

Um Probleme mit Ihren Rechenknoten zu beheben, müssen Sie möglicherweise Informationen über die VPC, das Subnetz und die Sicherheitsgruppen abrufen, die Ihren Instances zugeordnet sind. Wenn Sie Ihre Instance-IDs nicht kennen, finden Sie weitere Informationen unter Suchen nach Rechenknotengruppeninstanzen in AWS PCS.

AWS-Managementkonsole

Um VPC, Subnetz und Sicherheitsgruppen abzurufen

1. Öffnen Sie die Amazon EC2-Konsole.
2. Wählen Sie Instances.
3. In der Instanzen Tabelle, wählen Sie die Instanz-ID.
4. Finden Sie die VPC-ID und Subnetz-ID in der angezeigten Instanzübersicht für die Instanz.
5. Wählen Sie in der Instanzübersicht die Sicherheit Tab.
6. Finde den Sicherheitsgruppen in der Sicherheit Tab.

AWS CLI

Verwenden Sie den folgenden Befehl, um VPC-, Subnetz- und Sicherheitsgruppeninformationen für Ihre Instance abzurufen:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0 --query
 'Reservations[*].Instances[*].
{InstanceId:InstanceId,VpcId:VpcId,SubnetId:SubnetId,SecurityGroups:SecurityGroups[*].GroupI
 --output table
```

Probleme bei der Node-Registrierung

Die Knotenregistrierung ist die erste Aktion, die von einem Rechenknoten während des Bootstraps ausgeführt wird. Der Knoten ruft den AWS PCS-API-Endpunkt, bei dem man sich registrieren soll AWS STCK. Bei Registrierungsfehlern werden in der Regel Fehlermeldungen angezeigt, die den folgenden ähneln:

```
<13>Nov 13 16:23:50 user-data: [2025-11-13T16:23:50.510+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Registering node to cluster <clusterId>
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.192+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Retriable exception detected.
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.193+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Response is [specific error message]
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.194+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Retrying in 31 seconds...
<13>Nov 13 16:24:18 user-data: [2025-11-13T16:24:18.192+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Retriable exception detected.
...
<13>Nov 13 16:25:18 user-data: [2025-11-13T16:25:18.195+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Registration timeout (600 seconds) reached. Exiting.
<13>Nov 13 16:25:18 user-data: [2025-11-13T16:25:18.200+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: ERROR: Error: (2) occurred on line 1 when running /opt/aws/pcs/
bin/pcs_bootstrap_init.sh. Shutting down instance.
```

Falsches Instanzprofil

Wenn sich der Knoten aufgrund eines falschen Instanzprofils nicht registrieren kann, wird der folgende Fehler angezeigt:

```
<13>Nov 13 18:43:08 user-data: [2025-11-13T18:43:08.268+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_init.sh: INFO: Response is {
```

```
<13>Nov 13 18:43:08 user-data:   "__type":
 "com.amazon.coral.service#AccessDeniedException",
<13>Nov 13 18:43:08 user-data:   "Message": "User: arn:aws:sts::<accountId>:assumed-
role/<roleName>/<instanceId> is not authorized to perform:
 pcs:RegisterComputeNodeGroupInstance on resource:
 arn:aws:pcs:<regionCode>:<accountId>:cluster/<clusterId> as either the resource does
 not exist, some policy explicitly denies access, or no policy grants access",
<13>Nov 13 18:43:08 user-data:   "nodeID": null
<13>Nov 13 18:43:08 user-data: }
```

Vergewissern Sie sich, dass das dem Rechenknoten zugeordnete Instanzprofil das `pcs:RegisterComputeNodeGroupInstance` Erlaubnis. Weitere Informationen zum Erstellen eines gültigen Instanzprofils finden Sie unter Erstellen Sie ein Instanzprofil für AWS STCK.

Es kann keine Verbindung hergestellt werden AWS PCS-Endpunkte

Wenn sich Ihre Rechenknoten in einem privaten Subnetz befinden, stellen Sie sicher, dass Sie VPC-Endpunkte für konfiguriert haben AWS STCK. Stellen Sie alternativ sicher, dass Ihr Subnetz eine Route zu einem NAT-Gateway für den Internetzugang hat. In der Standardeinstellung ist der AWS Der PCS-Agent verwendet den Dual-Stack-Endpunkt ohne FIPS `pcs.region.api.aws`. Wenn Sie benutzerdefiniertes DNS verwenden, stellen Sie sicher, dass es aufgelöst werden kann `pcs.region.api.aws` zum Endpunkt. Weitere Informationen finden Sie hier:

- Greifen Sie auf ein AWS Dienst, der einen Schnittstellen-VPC-Endpunkt verwendet in der Virtuelle private Cloud von Amazon AWS PrivateLink Anleitung.
- Endpunkte und Servicekontingente für AWS 5 STÜCK.
- Verbinden Sie Ihre VPC mit anderen Netzwerken in der Amazon Virtual Private Cloud-Benutzerhandbuch
- AWS PCS-Netzwerke

Falsch konfiguriert AWS PCS-Endpunkt

Wenn Sie eine Fehlermeldung ähnlich der folgenden sehen, überprüfen Sie die mit Ihrem AWS PCS-VPC-Endpunkt:

```
com.amazon.coral.security.AccessDeniedException: User: arn:aws:sts::xxx:assumed-
role/<roleName>/<instanceId> is not authorized to perform:
 pcs:RegisterComputeNodeGroupInstance on resource:
```

```
arn:aws:pcs:<regionCode>:<accountId>:cluster/<clusterId> as either the resource does
not exist, some policy explicitly denies access, or no policy grants access
```

Weitere Informationen zur Konfiguration von VPC-Schnittstellenendpunkten für AWS PCS, siehe Zugriff AWS Parallel Computing Service mithilfe eines Schnittstellenendpunkts (AWS PrivateLink).

Instanz in einem öffentlichen Subnetz ohne öffentliche IP

Wenn Ihr Subnetz nicht öffentliche IP automatisch zuweisen aktiviert und Ihre Routenkonfiguration verwendet ein Internet-Gateway. Instances können nicht mit dem kommunizieren AWS PCS-API.

Instanzen in einem Subnetz mit einem Internet-Gateway müssen eine öffentliche IP-Adresse haben. Um dieses Problem zu beheben, wählen Sie eine der folgenden Optionen:

- Fügen Sie einen VPC-Endpunkt hinzu für AWS PCS zu Ihrer Cluster-VPC. Dadurch können Instances kommunizieren mit AWS PCS, ohne dass eine öffentliche IP-Adresse das Internet-Gateway passieren muss.
- Verwenden Sie ein privates Subnetz mit einem NAT-Gateway, sodass keine öffentliche IP-Adresse erforderlich ist.
- Aktivieren Sie die automatische Zuweisung öffentlicher IP-Adressen über Ihr Subnetz oder starten Sie die Vorlage, damit Instances die API über das Internet-Gateway kontaktieren können. Beachten Sie, dass diese Option für Interface-Instances mit mehreren Netzwerken nicht gültig ist.

Multi-NIC Instanz in einem öffentlichen Subnetz

Sie müssen ein privates Subnetz verwenden, wenn Sie einen Instance-Typ verwenden, der über mehrere Netzwerkschnittstellen (NICs) verfügt.

AWS Öffentliche IP-Adressen können nur Instances zugewiesen werden, die mit einer einzigen Netzwerkschnittstelle gestartet wurden. Weitere Informationen zu IP-Adressen finden Sie unter Weisen Sie beim Start der Instanz eine öffentliche IPv4-Adresse zu in der Amazon EC2-Benutzerhandbuch für Linux-Instances.

Multi-NIC Instanztypen benötigen ein NAT-Gateway oder einen internen Proxy im Subnetz für den Zugriff auf AWS PCS-Endpunkt. Alternativ können Sie einen VPC-Endpunkt hinzufügen für AWS PCS zu Ihrer Cluster-VPC.

Der geheime Clusterschlüssel wurde gelöscht oder zum Löschen markiert

Wenn der Slurm das Geheimnis geteilt hat AWS Secrets Manager wurde gelöscht oder zum Löschen markiert, Rechenknoten können sich nicht registrieren und Ihr Cluster wird beeinträchtigt.

AWS PCS erstellt automatisch ein gemeinsames Slurm-Geheimnis in AWS Secrets Manager (mit Namensformat: `pcs!slurm-secret-<cluster-id>`), wenn Sie einen Cluster erstellen. Dieses Geheimnis ist für die sichere Kommunikation im Cluster erforderlich. Weitere Informationen finden Sie unter Arbeiten mit Cluster-Secrets in AWS STCK.

Wenn dieses Geheimnis gelöscht oder zum Löschen markiert wird, können keine neuen Knoten dem Cluster und dem Controller oder anderen Cluster-Daemons beitreten (wie `slurmd` und `slurmdbd`) kann dem Cluster möglicherweise nicht wieder beitreten, wenn es neu gestartet wird.

Um dieses Problem zu beheben, können Sie das gelöschte Geheimnis wiederherstellen, sofern es sich noch innerhalb des Wiederherstellungsfensters befindet. Eine ausführliche Anleitung finden Sie unter Stellen Sie eine wieder her AWS Geheim des Secrets Managers.

Wenn das Wiederherstellungsfenster abläuft, kann das Geheimnis nicht wiederhergestellt werden und die betroffenen AWS Der PCS-Cluster kann nicht wiederhergestellt werden. Sie müssen einen neuen Cluster mit derselben Konfiguration erstellen. AWS PCS erstellt automatisch ein neues Scheduler-Geheimnis.

Probleme beim Verbinden des Slurm-Clusters

Nach erfolgreicher Knotenregistrierung versucht der Rechenknoten, dem Slurm-Cluster beizutreten. Der `slurmd` Der Daemon auf dem Knoten kontaktiert den Slurm-Controller, um sich beim Cluster zu registrieren. Bei Slurm-Join-Fehlern werden normalerweise Fehlermeldungen ähnlich den folgenden angezeigt:

```
<13>Nov  5 17:20:29 user-data: [2024-11-05T17:20:28+00:00] FATAL:
 Mixlib::ShellOut::ShellCommandFailed: service[slurmd] (aws-pcs-slurm::finalize_slurm
 line 18) had an error: Mixlib::ShellOut::ShellCommandFailed: Expected process to exit
 with [0], but received '1'
<13>Nov  5 17:20:29 user-data: ---- Begin output of ["/usr/bin/systemctl", "--system",
 "start", "slurmd"] ----
<13>Nov  5 17:20:29 user-data: STDOUT:
<13>Nov  5 17:20:29 user-data: STDERR: Job for slurmd.service failed because the
 control process exited with error code. See "systemctl status slurmd.service" and
 "journalctl -xe" for details.
```

```
<13>Nov  5 17:20:29 user-data: ---- End output of ["/usr/bin/systemctl", "--system",
 "start", "slurmd"] ----
```

Sicherheitsgruppenkonfiguration

Stellen Sie sicher, dass Ihre Sicherheitsgruppen korrekt konfiguriert sind, um die Kommunikation zwischen Rechenknoten und dem Slurm-Controller zu ermöglichen. Die Sicherheitsgruppen müssen den folgenden Verkehr zulassen:

- Port 6817 für `slurmd` um zu kommunizieren mit `slurmctld`
- Port 6818 für `slurmctld` zu pingen `slurmd`

Weitere Informationen zu den Anforderungen von Sicherheitsgruppen finden Sie in den folgenden Themen:

- Sicherheitsgruppen für AWS PCS erstellen
- Erstellen Sie Startvorlagen für AWS 5 STÜCK
- Anforderungen und Überlegungen zur Sicherheitsgruppe

> **Important**
>
> Die Cluster-Sicherheitsgruppe, die Sie Ihrem Cluster bei der Clustererstellung zugeordnet haben, muss auch in den Sicherheitsgruppen Ihrer Rechenknotengruppe konfiguriert werden, damit Rechenknoten mit dem Controller kommunizieren können.

Fehlende NVIDIA-Treiber

Wenn die Instance korrekt bootet, Jobs aber nicht starten und Sie in Ihren Instance-Protokollen Fehlermeldungen ähnlich den folgenden sehen, fehlen Ihnen möglicherweise NVIDIA-Treiber:

```
<13>Dec  2 13:52:00 user-data: [2024-12-02T13:52:00.094+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_config_always.sh: INFO: nvidia-smi not found!
...
<13>Dec  2 13:54:10 user-data: Job for slurmd.service failed because the control
 process exited with error code. See "systemctl status slurmd.service" and "journalctl
 -xe" for details.
```

```
<13>Dec  2 13:54:12 user-data: [2024-12-02T13:54:12.718+00:00] - /opt/aws/pcs/bin/
pcs_bootstrap_finalize.sh: INFO: systemctl could not start slurmd!
```

Wenn Sie eine Verbindung zur Instance herstellen und das überprüfen `slurmd` Daemon-Status, möglicherweise wird ein Fehler ähnlich dem folgenden angezeigt:

```
$ systemctl status slurmd
...
fatal: can't stat gres.conf file /dev/nvidia0: No such file or directory
```

Um dieses Problem zu beheben, installieren Sie NVIDIA-Treiber auf Ihrem benutzerdefinierten AMI. Weitere Informationen finden Sie unter Schritt 4 — (Optional) Zusätzliche Treiber, Bibliotheken und Anwendungssoftware installieren.

ResumeTimeout erreicht

Wenn ein Rechenknoten und seine EC2-Instance beendet werden, weil der Knoten defekt ist, AWS PCS unterstützt das AMI möglicherweise nicht oder es liegen Netzwerkprobleme vor. Die EC2-Instance läuft ungefähr 30 Minuten lang, bis Slurm ResumeTimeout ist erreicht und markiert den Knoten als `DOWN`.

Wenn die Instanz nicht korrekt bootet und nicht registriert ist AWS PCS (nein `RegisterComputeNodeGroupInstance` rufen Sie die EC2-Instance auf), überprüfen Sie Ihre Instance-Protokolle auf Fehlermeldungen, die den folgenden ähneln:

```
/opt/aws/pcs/bin/pcs_bootstrap_init.sh: No such file or directory
```

Dieser Fehler weist darauf hin, dass AWS Die PCS-Bootstrap-Software ist nicht Teil des AMI. Um dieses Problem zu beheben, stellen Sie sicher, dass Ihr benutzerdefiniertes AMI Folgendes enthält AWS PCS-Bootstrap-Software. Weitere Informationen finden Sie unter Benutzerdefinierte Amazon Machine Images (AMIs) für AWS PCS.

Slurmctld kann den Rechenknoten nicht pingen

Wenn die Instanz die Bootstrap-Prozedur korrekt ausführt und bei registriert ist AWS PCS, aber `slurmctld` kann sie nicht sehen und keine Jobs an sie senden. Die Instanz ist auf eingestellt `DOWN` nach einiger Zeit und dann beendet.

Dies kann durch falsch konfigurierte Sicherheitsgruppen verursacht werden. Zum Beispiel, wenn Port 6817 aktiviert ist, um Folgendes zuzulassen `slurmd` um zu kommunizieren `slurmctld`, aber Port 6818 fehlt zum Zulassen `slurmctld` um zu pingen `slurmd`.

Stellen Sie sicher, dass Ihre Sicherheitsgruppen alle erforderlichen Regeln enthalten, wie unter Anforderungen und Überlegungen zur Sicherheitsgruppe.

Beheben Sie Fehler bei ungültigen Knotenregistrierungen (INVALID_REG) in AWS STCK

Wenn sich ein Knoten bei einer Hardware registriert, die mit seiner Konfiguration in Konflikt steht, setzt Slurm den Knotenstatus auf `INVALID_REG` und entleert es. Die des Knotens `Reason` Feld im `scontrol show node` Die Ausgabe meldet die Ursache. Die häufigsten Ursachen sind eine konfigurierte CPU-Topologie, die nicht zur Hardware passt, und bei aktiver GPU-Autoerkennung ein Sockellayout, das nicht zur erkannten GPU-Affinität passt. Für weitere Informationen darüber, wie AWS PCS konfiguriert die Topologie auf jeder Slurm-Version, siehe Konfiguration der Hardwaretopologie in AWS STCK.

Die CPU-Topologie entspricht nicht der Hardware

Ein konfiguriertes Socket-Layout, das mehr Ressourcen deklariert, als der Knoten meldet, führt dazu, dass der Knoten registriert wird, und zwar aus dem folgenden Grund:

```
NodeName=my-cng-1-1 ... State=DOWN+CLOUD+DRAIN+INVALID_REG
   Reason=Low socket*core*thread count [slurm@2026-09-07T11:45:03]
```

Der Text zur Begründung ist wortwörtlich; die verglichenen Zahlen erscheinen nur in `slurmctld` auf Debug-Protokollebene protokollieren, als `Node my-cng-1-1 has low socket*core*thread count (48 < 96)`. Das bedeutet die konfigurierte `Sockets` und `CoresPerSocket` Die Kombination übersteigt die tatsächliche Anzahl der Kerne des Knotens. Zum Beispiel ein `Sockets` Eine benutzerdefinierte Einstellung, die größer ist als die tatsächliche Anzahl der NUMA- oder L3-Cache-Domänen, führt zu einer Gesamtsumme, die die Hardware nicht erfüllen kann.

Die GPU-Affinität passt nicht zu den Socket-Grenzen

Wenn die automatische GPU-Erkennung aktiviert ist, führt ein Socket-Layout, das nicht zur Hardware passt, zur Registrierung des Knotens, und zwar aus dem folgenden Grund:

```
Reason=gres/gpu GRES autodetected core affinity 0-23 on node my-cng-1-1 doesn't match
 socket
boundaries. (Socket 0 is cores 0-0). Consider setting Parameters=l3cache_as_socket ...
```

Das bedeutet, dass die GPU an eine Reihe von Kernen angeschlossen ist, die nicht in einen konfigurierten Socket passen. Es tritt normalerweise auf, wenn die automatische GPU-Erkennung ohne CPU-Topologie aktiviert ist, sodass der Knoten auf das Standardlayout mit einem Kern pro Socket zurückfällt.

Auflösung

Konfigurieren Sie die CPU-Topologie der Rechenknotengruppe so, dass die Socket-Grenzen der Hardware entsprechen:

- Stellen Sie auf Slurm 25.11 `Sockets` (und `Parameters`) in den benutzerdefinierten Slurm-Einstellungen der Compute-Knotengruppe.
- In Slurm 26.05 und höher entferne oder korrigiere einen `Sockets` überschreiben, das nicht zur Hardware passt, und lass AWS PCS wenden die automatische Topologie an.

Weitere Informationen finden Sie unter Topologieverhalten nach Slurm-Version.

Behebung von Fehlern bei der Auftragsübermittlung aufgrund von MaxJobCount Begrenzung

Problem: Die Einreichung von Aufträgen schlägt mit der folgenden Fehlermeldung fehl:

```
sbatch: error: Slurm temporarily unable to accept job, sleeping and retrying
```

Dieser Fehler tritt auch dann auf, wenn die Anzahl der laufenden und ausstehenden Jobs deutlich unter dem Joblimit des Clusters zu liegen scheint.

Ursache: Der `MaxJobCount` Das Limit beinhaltet alle Jobs, die von Slurm verfolgt werden, nicht nur laufende oder ausstehende Jobs. Abgeschlossene Jobs bleiben für einen bestimmten Zeitraum (standardmäßig 5 Minuten) im Speicher von Slurm, bevor sie gelöscht werden. In Zeiten mit hohem Auftragsdurchsatz kann die Gesamtzahl der aktiven und der kürzlich abgeschlossenen Jobs den Grenzwert überschreiten.

Sie können die Gesamtzahl der Jobs überprüfen, indem Sie den folgenden Befehl auf einem Clusterknoten ausführen:

```
scontrol show jobs | grep -c JobId
```

Dies zeigt die Gesamtzahl der Jobs, die Slurm verfolgt, einschließlich abgeschlossener Jobs, die noch gelöscht werden müssen.

Lösung: Erwägen Sie einen der folgenden Ansätze:

- Erstellen Sie einen größeren Cluster — Wenn Ihre Arbeitslast ständig mehr gleichzeitige Jobs erfordert, erstellen Sie einen neuen Cluster mit einer größeren Größe. Weitere Informationen zu Clustergrößen und ihren Grenzwerten finden Sie unter Clustergröße in AWS PCS.
- Reduzieren Sie die Rate der Stellenausschreibungen — Passen Sie Ihre Job-Skripte so an, dass Jobs langsamer eingereicht werden, sodass die Zeit für abgeschlossene Jobs aus dem Tracking von Slurm gelöscht wird.

Behebung fehlender Jobdatensätze in Slurm Buchhaltung

Aufträge, die abgeschlossen wurden, fehlen in Slurm Buchhaltungsberichte, z. B. die Ausgabe der `sacct` oder `sreport` Befehle. Aufzeichnungen fehlen nur für ein begrenztes Zeitfenster, das in der Regel auf einen Zeitraum folgt, in dem die Zahl der eingereichten Stellen sehr hoch ist. Buchhaltungsberichte sind vor und nach diesem Zeitfenster vollständig. Dieses Thema bezieht sich auf Cluster mit Slurm Buchhaltung aktiviert. Weitere Informationen zur Buchhaltung finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.

Häufige Ursache

Das Slurm Controller-Daemon (`slurmctld`) schreibt Buchhaltungsdaten nicht direkt in die Buchhaltungsdatenbank. Es sendet die Daten an die Slurm Datenbank-Daemon (`slurmdbd`) über eine interne, speicherinterne Warteschlange. Die Warteschlange kann ihre maximale Größe erreichen, wenn Jobs schneller als eingereicht werden `slurmdbd` überträgt Datensätze in die Datenbank. Wenn die Warteschlange voll ist, `slurmctld` verwirft neue Buchhaltungsnachrichten, anstatt sie der Warteschlange hinzuzufügen.

Verworfene Nachrichten werden nicht erneut versucht, daher fehlt der Auftragsverlauf, den sie übertragen haben, dauerhaft in der Buchhaltungsdatenbank. Nur Buchhaltungsdaten sind betroffen. Die Auftragsplanung und die Auftragsausführung werden normal fortgesetzt. Zeichnet das auf

`slurmdbd` bereits festgeschriebene Daten verbleiben in der Datenbank. Die Abrechnung wird automatisch wieder aufgenommen, wenn die Warteschlange leer ist.

Auflösung

Um die Ursache zu bestätigen und die Wahrscheinlichkeit eines erneuten Auftretens zu verringern

1. Suchen Sie nach `slurmctld` Protokoll des Clusters nach Einträgen ähnlich den folgenden.

   ```
   error: agent queue is full (33794), discarding DBD_JOB_START:1425 request
   ```

   Jeder Eintrag entspricht einer verworfenen Buchhaltungsnachricht. Der Nachrichtentyp identifiziert den Datensatz, der verloren gegangen ist, z. B. `DBD_JOB_START`, `DBD_JOB_COMPLETE`, `DBD_STEP_START`, oder `DBD_STEP_COMPLETE`. Wenn der Cluster keine Scheduler-Protokolle übermittelt, richten Sie die Protokollzustellung ein. Auf diese Weise können Sie die Ursache überprüfen, falls der Zustand erneut auftritt. Weitere Informationen finden Sie unter Scheduler loggt sich in AWS PCS ein.
2. Stellen Sie das ein `CommitDelay` Parameter auf 1 in der slurmdbd-Konfiguration des Clusters. Mit dieser Einstellung `slurmdbd` gruppiert Datenbank-Commits, anstatt jeden Datensatz einzeln festzuschreiben, was die Geschwindigkeit erhöht, mit der die Warteschlange entleert wird. Verwenden Sie den `SlurmdbdCustomSettings` Eigenschaft des Clusters, um die Einstellung anzuwenden. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter SlurmDBD-Einstellungen in AWS 5 STÜCK und CommitDelay in der Slurm Dokumentation.

   > **Important**
   >
   > `SlurmdbdCustomSettings` ersetzt die slurmdbd-Einstellungen des Clusters, anstatt sie zu ergänzen. Jede Einstellung, die Sie in der Aktualisierungsanforderung weglassen, wird entfernt. Wenn der Cluster möglicherweise bereits slurmdbd-Einstellungen hat, führen Sie Folgendes aus `aws pcs get-cluster` fügen Sie zuerst hinzu `{parameterName=CommitDelay,parameterValue="1"}` zur abgerufenen Liste und dann die vollständige Liste absenden.

   Example— Einstellung CommitDelay auf einem Cluster

   ```
   aws pcs update-cluster --cluster-identifier my-cluster \
   --slurm-configuration \
   ```

```
'SlurmdbdCustomSettings=[{parameterName=CommitDelay,parameterValue="1"}]'
```

3. Vergewissern Sie sich, dass die Einstellung vorhanden ist. In der AWS-Managementkonsole, anzeigen Zusätzliche Scheduler-Einstellungen für den Cluster. In der AWS CLI, führen Sie den folgenden Befehl aus und überprüfen Sie `slurmConfiguration` Feld der Antwort.

```
aws pcs get-cluster --cluster-identifier my-cluster
```

> **Note**
>
> Die Buchhaltung zeichnet das auf `slurmctld` verworfene können nicht wiederhergestellt werden. Jobs, die während des betroffenen Zeitfensters ausgeführt wurden, fehlen in den Buchhaltungsberichten.

Prävention

Um zu erkennen, ob die Warteschlange überlastet ist, und nicht erst später nach fehlenden Datensätzen suchen zu müssen, senden Sie `slurmctld` loggt sich bei Amazon ein CloudWatch Loggt, Amazon Simple Storage Service (Amazon S3) oder Amazon Data Firehose und überwacht sie auf `agent queue is full` Einträge. Weitere Informationen finden Sie unter Scheduler loggt sich in AWS PCS ein.

Behalten `CommitDelay` eingestellt auf 1 auf Clustern, auf denen Workloads mit einer hohen Rate an Auftragsübermittlungen ausgeführt werden.

Dokumenthistorie für den AWS PCS-Benutzerhandbuch

In der folgenden Tabelle werden die wichtigen Änderungen an der Dokumentation für beschrieben AWS STCK.

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
28. Oktober 2026	AWS PCS wurde in Kanada (Central) veröffentlicht	AWS PCS ist jetzt in Kanada (Central) (ca-central-1) erhältlich. CloudFormation Vorlagen sind für den Einstieg in Kanada (Central) verfügbar AWS-Regio n. Weitere Informati onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend
15. Oktober 2026	AWS PCS wurde im asiatisch-pazifischen Raum (Hyderabad) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Hyderabad) (ap-south-2) erhältlich. CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifisc hen Raum (Hyderabad) verfügbar AWS-Regio n. Weitere Informati	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	
15. Oktober 2026	Option 1: Fortlaufendes Update: AMIs mit mehreren Versionen werden jetzt unterstützt	Das Verfahren für Option 1 (fortlaufendes Update) wurde aktualisiert, um AMIs zu unterstützen, die die aktuelle Version enthalten Slurm Version und eine oder mehrere Zielversionen sowie zur Klärung, wie Rechenknoten, auf denen noch die Vorgängerversion ausgeführt wird, ersetzt werden können. Weitere Informationen finden Sie unter Aktualisieren Sie die Scheduler-Version eines AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
1. Oktober 2026	AWS PCS im asiatisch-pazifischen Raum (Malaysia) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Malaysia) erhältlich (ap-southeast-5). CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifischen Raum (Malaysia) verfügbar AWS-Region. Weitere Informationen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend
30. September 2026	Von AWS verwaltete Aktualisierung der Richtlinien	AWS PCS hat das aktualisiert AWSPCSServiceRolePolicy zur Unterstützung mehrerer Netzwerkschnittstellen für Recheninstanzen. Weitere Informationen finden Sie unter AWS verwaltete Richtlinien für AWS Dienst für parallele Datenverarbeitung.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
11. September 2026	Unterstützung für Slurm 26.05, benutzerdefinierte GRES-Einstellungen und Hardwaretopologie	AWS PCS unterstützt Slurm 26.05. Cluster, auf denen 25.11, 25.05 und 24.11 ausgeführt wird, können direkt auf 26.05 aktualisiert werden. Ein Einzelbeispiel-AMI unterstützt jetzt alle Slurm Version, bei der es sich nicht um EOL handelt, und der AMI-Name enthält keine Slurm Version. Weitere Informationen erhalten Sie unter Slurm-Versionen in AWS 5 STCK und Versionshinweise für Slurm-Versionen in AWS 5 STCK. Das MIG-Thema listet jetzt auf, was ein MIG-Knoten bei jedem Start benötigt, und es wird eine Aktion für den Knotenlebenszyklus empfohlen, die ihn anwendet und den Knoten fehlschlägt, wenn der MIG-Modus nicht aktiviert werden kann. Weitere Informationen finden Sie unter Konfiguration Multi-Instance GPU (MIG) in AWS STCK.	2026-09-09

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		Sie können konfiguri eren Slurm Generisch e Ressourceneinträge (GRES) für eine Rechenknotengruppe mit dem `gresCusto mSettings` Eigenscha ft, und deklarieren Sie das planbare GRES mit dem `Gres` kundenspe zifisch Slurm Einstellu ng. AWS PCS aktiviert standardmäßig auch die automatische GRES-Erke nnung auf Rechenkno tengruppen, die über GPUs verfügen. Weitere Informationen finden Sie unter Konfigura tion benutzerdefinierter GRES-Einstellungen in AWS 5 STCK. Der Slurm Die REST-API liest den Benutzern amen aus dem `name` Feld innerhalb der `id` Behauptung eines JWT. Die oberste Ebene `sun`, `username`, und benutzerdefiniert `userclaimfield` Ansprüche sind veraltet Slurm 26.05 und später.	

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		Weitere Informationen finden Sie unter Authentifizierung mit der Slurm-REST-API in AWS 5 STCK. Es wurde ein Thema hinzugefügt, das beschreibt, wie Slurm modelliert CPU-Topologie und GPU-Affinität, was AWS PCS konfiguriert auf jedem Slurm Version und wie das Socket-Layout einer Compute-Knotengruppe konfiguriert wird. Ein Thema zur Problembehandlung für `INVALID_REG` Node-Registrierungen sind ebenfalls verfügbar. Weitere Informationen finden Sie unter Konfiguration der Hardwaretopologie in AWS STCK. Es wurde ein Thema hinzugefügt, das beschreibt, wie NVIDIA-GPUs partitioniert werden Multi-Instance GPU-Instances (MIG) und Planung dieser Instances als GRES-Typen in einer	

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		Rechenknotengruppe , einschließlich der bekannten Einschrän kungen für ausgescha ltete Knoten. Weitere Informationen finden Sie unter Konfiguration Multi-Instance GPU (MIG) in AWS STCK.	
10. September 2026	AWS PCS wurde im asiatisch-pazifischen Raum (Melbourne) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Melbourne) (ap-southeast-4) erhältlich. CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifisc hen Raum (Melbourne) verfügbar AWS-Regio n. Weitere Informati onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
7. September 2026	Bereitstellung von Clusterdetails und Herstellen einer Verbindung zu mehreren Clustern	Das Thema zum Konfigurationsskript für den Anmeldeknoten wurde aktualisiert und beschreibt nun, wie Clusterdetails in der Befehlszeile angegeben werden, sodass der Befehl übersprungen wird AWS PCS-API-Aufruf. Außerdem wurde eine Anleitung zum Verbinden eines Anmeldeknotens mit mehreren hinzugefü gt AWS PCS-Cluster, einschließlich Cluster in verschiedenen AWS-Regionen.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
3. September 2026	AWS PCS wurde im asiatisch-pazifischen Raum (Neuseeland) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Neuseeland) erhältlich (ap-south east-6). CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifischen Raum (Neuseeland) verfügbar AWS-Regio n. Weitere Informati onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend
20. August 2026	Die Slurm-REST-API-Dokumentation wurde korrigiert	Das Beispielskript zum Abrufen des zu verwendenden JWT-Signaturschlüssels wurde korrigiert `--version-id` statt `--version-stage` , und aktualisierte Endpunkt-URLs in den Slurm-REST-API-Themen zur Verwendung `http://`, weil `slurmrestd` Endpunkt ist nur HTTP.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
14. August 2026	Die bewährten Methoden für Aktionen im Knotenlebenszyklus wurden aktualisiert	Es wurde eine Anleitung hinzugefügt, die AWS PCS unterstützt nicht den Neustart der Instance während eines Lifecycle-Aktionsskripts.	Nicht zutreffend
11. August 2026	Aktualisiert AWS PCS-Agent auf Version 1.5.1	Hinzugefügt AWS PCS-Agent-Version 1.5.1-1 im Benutzerhandbuch.	Nicht zutreffend
11. August 2026	Dokumentierte Beschränkung der fortlaufenden Updates für Slurm CLI Filter Plugins	Der Aktualisierungsseite für die Cluster-Version wurde eine Einschränkung hinzugefügt: Cluster, die Slurm CLI Filter Plugins verwenden, unterstützen keine fortlaufenden Updates.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
10. August 2026	Neues Thema: Fehlende Jobdatensätze in Slurm Buchhaltung	Es wurde ein Thema zur Problembehandlung hinzugefügt, das beschreibt, warum Jobs in den Slurm-Buchhaltungsberichten nach einer Zeit mit hoher Auftragseinreichungsrate fehlen können, wie man die Ursache im Slurm-Controller-Log bestätigt und wie man das einstellt `CommitDelay` slurmdbd-Parameter für einen Cluster. Weitere Informationen finden Sie unter Behebung fehlender Jobdatensätze in Slurm Buchhaltung.	Nicht zutreffend
6. August 2026	Das Tutorial „Erste Schritte“ verwendet Aktionen im Knotenlebenszyklus	Das Tutorial für die ersten Schritte wurde aktualisiert, sodass Compute-Knotengruppen gemeinsam genutzten Speicher bereitstellen und Knotenprotokolle mit Node-Lifecycle-Aktionen weiterleiten, anstatt Benutzerdaten von Startvorlagen zu starten.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
6. August 2026	Neues Thema: Sammle Slurm-Metriken mit einem verwalteten Prometheus-Sammler	Eine aufgabenorientierte Anleitung zur Konfiguration eines verwalteten Prometheus-Sammlers zum Scrapen von Slurm hinzugefügt OpenMetrics von einem AWS PCS-Cluster-Controller.	Nicht zutreffend
29. Juli 2026	Aktualisierte Slurm-Installationsprogramme	Das Benutzerhandbuch für Slurm-Installateure wurde aktualisiert. Version 25.11.7-1 hinzugefügt.	Nicht zutreffend
28. Juli 2026	Die Beschreibung von Capacity Blocks for ML wurde aktualisiert	Die Beschreibung der Amazon EC2-Kapazitätsblöcke für ML auf der Seite „Kapazitätsreservierungen“ wurde aktualisiert.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
28. Juli 2026	Die Dokumentation zu Aktionen im Knotenleb enszyklus wurde aktualisi ert	Eine Empfehlung, die nicht unterstützt wird, wurde von der Seite mit Aktionen zum Knotenleb enszyklus entfernt. Dem überprüften Skript configure-cloudwatch-logs wurde ein Hinweis hinzugefügt, der klarstell t, dass das Skript das bestehende Amazon beibehält CloudWatch Agentenkonfiguration.	Nicht zutreffend
23. Juli 2026	Dokumentieren Sie die Einschränkungen der QOS-Warteschlangen einstellungen	Es wurde eine Beschränk ung der QOS-Existenz in der Kontoführungsdaten bank für QOS-Warte schlangeneinstellungen hinzugefügt.	Nicht zutreffend
23. Juli 2026	Aktualisiert AWS PCS-Agent auf Version 1.5.0	Hinzugefügt AWS PCS-Agent-Version 1.5.0-1 im Benutzerhandbuch. Diese Version bietet Unterstüt zung für Aktionen im Knotenlebenszyklus.	2026-07-23

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
31. Juli 2026	Neues Thema: Slurm-Konfiguration	Es wurde ein Thema hinzugefügt, das beschreibt, wie AWS PCS verwaltet die Slurm-Konfiguration eines Clusters im Slurm-Modus ohne Konfiguration. In diesem Modus erhalten Rechenknoten und Anmeldeknoten ihre Konfiguration vom Cluster-Controller. Weitere Informationen finden Sie unter Slurm-Konfiguration in AWS STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
20. Juli 2026	AWS PCS wurde im asiatisch-pazifischen Raum (Seoul) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Seoul) (ap-northeast-2) erhältlich. CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifischen Raum (Seoul) verfügbar AWS-Region. Weitere Informationen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
15. Juli 2026	Die Slurm-Installation sprogramme wurden aktualisiert	Das Benutzerhandbuch für Slurm-Installateure wurde aktualisiert. Die Versionen 23.11.10-5, 24.05.8-3, 24.11.7-2, 25.05.8-1, 25.05.8-2, 25.11.6-1 und 25.11.6-2 wurden hinzugefü gt. AWS PCS hat die 24.05-Versionen als veraltet markiert. Weitere Informationen finden Sie unter Softwareinstallati onsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK.	Nicht zutreffend
14. Juli 2026	Option 1: Fortlaufendes Update: AMI mit einer einzigen Version ist jetzt ausreichend	Das Verfahren für Option 1 (fortlaufendes Update) wurde aktualisiert, um zu berücksichtigen, dass AMIs mit einer einzigen Version jetzt unterstützt werden. Weitere Informati onen finden Sie unter Aktualisieren Sie die Scheduler-Version eines AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
29. Juni 2026	Neue Funktion: Aktualisieren Sie die Scheduler-Version eines Clusters	Sie können jetzt die Scheduler-Version auf einer vorhandenen Version aktualisieren AWS PCS-Cluster, ohne Ihre Infrastruktur neu aufzubauen. Weitere Informationen finden Sie unter Aktualisierung der Scheduler-Version eines Clusters in AWS STCK. Für das entsprechende CloudFormation Ressource finden Sie unter `AWS::PCS::Cluster` in der AWS CloudFormation Benutzerleitfaden.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
19. Juni 2026	AWS PCS wurde in Afrika (Kapstadt) und im asiatisch-pazifischen Raum (Jakarta) veröffentlicht	AWS PCS ist jetzt in Afrika (Kapstadt) (af-south-1) und im asiatisch-pazifischen Raum (Jakarta) (ap-southeast-3) erhältlich. CloudFormation Vorlagen sind für den Einstieg in Afrika (Kapstadt) und im asiatisch-pazifischen Raum (Jakarta) verfügbar AWS-Region. Weitere Informationen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend
01. Juni 2026	Neues Thema: PCS-ready DLAMI	Dokumentation für das AWS hinzugefügt PCS-ready DLAMI Base GPU AMI, ein AWS-maintained AMI zum Laufen AI/ML und HPC-Workloads aktiviert AWS 5 STCK. Weitere Informationen finden Sie unter Verwenden von PCS-ready DLAMI mit AWS 5 STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
28. Mai 2026	Neue Funktion: Berechnung der Leerlaufzeit beim Herunterskalieren auf Knotengruppenebene	Einzelne Compute-Knotengruppen können jetzt die Leerlaufzeit beim Herunterskalieren auf Cluster-Ebene durch ihre eigene überschreiben `scaleDownIdleTimeInSeconds` Einstellung. Diese Funktion erfordert Slurm Version 25.11 oder höher.	Nicht zutreffend
26. Mai 2026	AWS PCS wurde im asiatisch-pazifischen Raum (Osaka) und in Südamerika (São Paulo) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Osaka) (ap-northeast-3) und in Südamerika (São Paulo) (sa-east-1) erhältlich. CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifischen Raum (Osaka) und Südamerika (São Paulo) verfügbar AWS-Regionen. Weitere Informationen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
18. Mai 2026	Neues Feature: Gemischte Skalierung gskapazität für Rechenknotengruppen	Sie können jetzt Rechenknotengruppen mit gemischter Skalierung konfigurieren, wobei die Mindestanzahl an Instanzen größer als Null und kleiner als die maximale Anzahl an Instanzen ist. Auf diese Weise wird ein Grundwert an ständig verfügbarer Kapazität mit dynamischem Scale-Out beibehalten. Diese Funktion erfordert Slurm 24.05 oder später. Weitere Informationen finden Sie unter Erstellen einer Compute-Knotengruppe in AWS STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
7. Mai 2026	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.4.0-1. Der AWS Der PCS-Agent unterstützt jetzt das Konfigurieren und Starten von `slurmd` Daemon für jede Slurm-Version, die mit dem auf dem Cluster laufenden Slurm-Controller kompatibe l ist. Weitere Informati onen erhalten Sie unter Softwareinstallati onsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend
29. April 2026	Das Installationsprogr amm für Slurm 25.05 wurde aktualisiert	Das AMI-Thema für den Slurm-Installer am 25.05.7-1 wurde aktualisiert. Weitere Informationen erhalten Sie unter Softwarei nstallationsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
23. April 2026	AWS PCS in Europa (Spanien) veröffentlicht	AWS PCS ist jetzt in Europa (Spanien) (EU-Süd-2) erhältlich. CloudFormation Vorlagen sind für den Einstieg in Europa (Spanien) verfügbar AWS-Regio n. Weitere Informati onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
23. APRIL 2026	Für Slurm 25.11 aktualisiert	Das Benutzerhandbuch für die Unterstützung von Slurm 25.11.2-1 wurde aktualisiert. Slurm 25.11 ist jetzt die Standardversion. Beispiel-AMIs für Slurm 25.11 basieren jetzt auf Amazon Linux 2023. Weitere Informationen finden Sie hier: • Slurm-Versionen in AWS 5 STCK • Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK • Versionshinweise für AWS PCS-Beispiel-AMIs	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
22. April 2026	Neue Funktion: Scheduler -Auditprotokolle	AWS PCS liefert die Auditprotokolle des Schedulers jetzt separat über `PCS_SCHEDULER_AUDIT_LOGS` Protokolltyp für Cluster, auf denen Slurm 25.11 oder höher ausgeführt wird. Weitere Informationen finden Sie unter Scheduler-Auditprotokolle in AWS PCS.	Nicht zutreffend
22. April 2026	Die Scheduler-Logs wurden mit slurmdbd und slurmrestd aktualisiert	AWS PCS liefert jetzt `slurmdbd` Logs (Slurm 24.11+) und `slurmrestd` loggt (Slurm 25.05+) durch das bestehende `PCS_SCHEDULER_LOGS` Protokolltyp. Weitere Informationen finden Sie unter Scheduler loggt sich in AWS PCS ein.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
16. April 2026	AWS PCS in Europa veröffentlicht (Mailand)	AWS PCS ist jetzt in Europa (Mailand) (eu-south-1) erhältlich. CloudFormation Vorlagen sind für den Einstieg in Europa (Mailand) verfügbar AWS-Regio n. Weitere Informati onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend
10. März 2026	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.3.2-1. Es wurde ein Problem behoben, das den Bootstrap für RHEL 8.10 und Rocky Linux 8.10 Compute Node betraf. Weitere Informati onen erhalten Sie unter Softwareinstallati onsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
11. Februar 2026	AWS PCS wurde im asiatisch-pazifischen Raum (Mumbai) und in Europa (Paris) veröffentlicht	AWS PCS ist jetzt im asiatisch-pazifischen Raum (Mumbai) (ap-south-1) und in Europa (Paris) (eu-west-3) erhältlich. CloudFormation Vorlagen sind für den Einstieg im asiatisch-pazifischen Raum (Mumbai) verfügbar AWS-Region und Europa (Paris) AWS-Region. Weitere Informationen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend
18. November 2025	Neue Funktion: Slurm REST API	Die Slurm REST API wird jetzt für Slurm 25.05 oder höher unterstützt. Weitere Informationen finden Sie unter Setzen Sie die REST-API ein AWS 5 STCK.	AWS-SDK: 18.11.2025

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
17. November 2025	Die Slurm-Installation sprogramme wurden aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS Slurm-Installation sprogramme 24.11.7-1 und 25.05.5-1. Weitere Informationen erhalten Sie unter Softwarei nstallationsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
10. November 2025	Neues Feature: Unterstützung für das Slurm-CLI-Filter-Plugin	AWS PCS unterstützt jetzt Slurm-CLI-Filter-Plugins, um benutzerdefinierte Lua-Skripte auszuführen, die die Parameter für die Auftragsübermittlung validieren und ändern, bevor sie den Slurm-Controller erreichen. Verwenden Sie CLI-Filter, um benutzerdefinierte Richtlinien durchzusetzen, Standardparameter festzulegen und die Benutzer bei der Auftragsübermittlung zu unterstützen. Für diese Funktion ist Slurm Version 25.05 oder höher erforderlich. Weitere Informationen finden Sie unter Verwenden Sie die Slurm CLI Filter Plugins, um die Auftragsübermittlung in anzupassen AWS 5 STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
7. November 2025	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.3.1-1. Weitere Informati onen erhalten Sie unter Softwareinstallati onsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend
3. November 2025	PCS-Agent und Slurm-Installationsprogramme wurden aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS Agent 1.3.0-1 und Slurm-Ins tallationsprogramme 24.11.6-2, 24.05.8-2 und 23.11.10-4. Die Liste der unterstützten Betriebssysteme wurde aktualisiert. Weitere Informationen erhalten Sie unter Softwarei nstallationsprogramme zum Erstellen benutzerd efinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
23. Oktober 2025	Aktualisierter Inhalt: pcs-multi-cluster-login-configure.sh	Einige Fehler im Konfigurationsskript für den Multi-Cluster-Anmeldeknoten wurden behoben. Weitere Informationen finden Sie unter AWS Konfigurationsskriptcode für den PCS-Multi-Cluster-Anmeldeknoten.	Nicht zutreffend
21. Oktober 2025	Neues Feature: Geheime Cluster-Rotation	AWS PCS unterstützt jetzt die Rotation geheimer Clusterschlüssel, um die Sicherheit zu erhöhen. Weitere Informationen finden Sie unter Rotierende Clustergeheimnisse in AWS PCS. Die minimalen Administratorberechtigungen wurden aktualisiert, um die Rotation geheimer Clusterschlüssel zu unterstützen. Weitere Informationen finden Sie unter Mindestberechtigungen für AWS 5 STÜCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
17. Oktober 2025	Neues Thema: Konfigurationsskript für Multi-Cluster-Anmeldeknoten	Es wurde ein neues Thema hinzugefügt, das ein Skript zur Konfiguration eines eigenständigen Anmeldeknotens für die Verbindung zu mehreren enthält AWS PCS-Cluster. Das Skript automatisiert die Konfiguration mehrerer Slurms `sackd` Daemons und erstellt Aktivierungsskripte für die Cluster-Interaktion. Weitere Informationen finden Sie unter Verbinden eines eigenständigen Anmeldeknotens mit mehreren Clustern in AWS STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
16. Oktober 2025	Für Slurm 25.05 aktualisiert	Das Benutzerhandbuch für die Unterstützung von Slurm 25.05 wurde aktualisiert. Slurm 25.05 ist jetzt die Standardversion. Weitere Informationen finden Sie hier: • Slurm-Versionen in AWS 5 STCK • Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK • Versionshinweise für AWS PCS-Beispiel-AMIs	Nicht zutreffend
16. Oktober 2025	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.2.2-1. Weitere Informationen erhalten Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK und AWS PCS-Agent-Versionen.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
2. Oktober 2025	Neue Funktionen: Neustart des Slurm-Knotens, Cluster-Updates und benutzerdefinierte Slurm-Einstellungen	AWS PCS bietet Unterstützung für mehrere neue Funktionen: • Neustart des Slurm-Knotens — Benutze die native Version von Slurm `scontrol reboot` Befehl zum Neustarten von Rechenknoten, ohne dass die Instanz ausgetauscht werden muss. Weitere Informationen finden Sie unter Rechenknoten mit Slurm neu starten AWS 5 STÜCK. • Cluster-Aktualisierungen — Ändern Sie Cluster-Konfigurationen nach der Erstellung ohne Neuerstellungen. Weitere Informationen finden Sie unter Aktualisierung eines Clusters in AWS STK.. • Benutzerdefinierte Slurm-Einstellungen — Konfigurieren Sie erweiterte Slurm-Parameter für Cluster-,	2025-10-01

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		Queue- und Compute Node Group-Ressourcen. Weitere Informationen finden Sie unter Konfiguration benutzerdefinierter Slurm-Einstellungen in AWS 5 STCK.	
23. September 2025	Neues Thema zur Problembehandlung: Bootstrap-Probleme mit Rechenknoten	Eine Anleitung zur Problembehandlung zur Diagnose und Behebung von Compute-Node-Bootstrap-Problemen wurde hinzugefügt. Weitere Informationen finden Sie unter Beheben Sie Probleme mit dem Bootstrap und der Registrierung von Rechenknoten in AWS STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
17. September 2025	Neues Feature: Kapazitätsblöcke für ML	AWS PCS unterstützt jetzt Amazon EC2-Kapazitätsblöcke für ML, mit denen Sie Reservierungen vornehmen können GPU-based beschleunigte Rechenins tanzen für Ihre Cluster. Weitere Informationen finden Sie unter Verwenden von Amazon EC2-Kapazitätsblöcken für ML mit AWS 2 STÜCK. Mindestberechtigungen zur Unterstützung von Kapazitätsblöcken sind jetzt Teil der Mindestberechtigungen für einen Dienstadministrator. Weitere Informationen finden Sie unter Mindestberechtigungen für AWS 5 STÜCK.	2025-09-17

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
11. Septemb 2025	Von AWS verwaltetes Richtlinien-Update	AWS PCS hat das aktualisiert AWSPCSServiceRolePolicy zur Unterstützung von Kapazitätsblöcken. Weitere Informationen finden Sie unter AWS verwaltete Richtlinien für AWS Dienst für parallele Datenverarbeitung.	Nicht zutreffend
14. August 2(	Die Dokumentation zum Instanzprofil wurde aktualisiert	Die Dokumentation zum Instanzprofil wurde um umfassende CLI-Anweisungen zum Erstellen von IAM-Rollen und Instanzprofilen erweitert. Schrittweise Anleitungen zum Einrichten von Instanzprofilen mithilfe von hinzugefügt AWS CLI und verbesserte Anleitung zum Auffinden von Instanzprofilen, die mit verwendet werden AWS 5 STCK. Weitere Informationen finden Sie unter IAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung.	14.08.2025

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
1. August 202	Neues Thema: SPANK-Plugins	Dokumentation für SPANK (Slurm) hinzugefügt Plug-in Plugins (Architecture for Node und Job Kontrol), mit denen Sie das Verhalten von Slurm beim Jobstart und bei der Ausführung von Jobs erweitern und ändern können AWS PCS-Cluster. Weitere Informationen finden Sie unter Erweitern Sie die Slurm-Funktionalität auf AWS PCS mit SPANK-Plugins.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
1. August 20	Unterstützung für IPv6-Netzwerke	Unterstützung für IPv6-Netzwerke beim Erstellen hinzugefügt AWS PCS-Cluster. Sie können jetzt IPv6 als Netzwerkt yp für Ihren Cluster auswählen, wobei die VPC-Anforderungen, die Subnetzkonfiguration, die Sicherheitsgruppen einstellungen und die Verfahren zur Clusterer stellung entsprechend aktualisiert werden. Weitere Informationen erhalten Sie unter AWS Anforderungen und Überlegungen zu PCS, VPC und Subnetzen und Einen Cluster erstellen in AWS STK..	2025-08-01

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
3. Juli 2025	AWS PCS wurde in Europa (London) veröffentlicht	AWS PCS ist jetzt in Europa (London) (eu-west-2) erhältlich. CloudFormation Vorlagen sind für den Einstieg in Europa (London) verfügbar AWS-Regio n. Weitere Informati onen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
1. Juli 2025	Die Anweisungen für die Konsole wurden aktualisiert	Du kannst jetzt haben AWS PCS erstellt ein grundlegendes Instanzprofil und eine Sicherheitsgruppe für Sie, wenn Sie in der Konsole einen Cluster und eine Rechenknotengruppe erstellen. Weitere Informationen finden Sie unter: • Einen Cluster erstellen in AWS STK. • Erstellen einer Compute-Knotengruppe in AWS STCK • IAM-Instanzprofile für AWS Dienst für parallele Datenverarbeitung	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
23. Juni 2025	Neue verwaltete Richtlinie: AWSPCSComputeNodePolicy	Es wurde eine neue verwaltete Richtlinie hinzugefügt, die folgende Berechtigungen gewährt AWS PCS-Rechenknoten, zu denen eine Verbindung hergestellt werden soll AWS PCS-Cluster. Weitere Informationen finden Sie unter AWS verwaltete Richtlinie: AWSPCSComputeNodePolicy.	Nicht zutreffend
19. Juni 2025	Neues Thema: Protokolle zur Fertigstellung von Aufträgen	Verwenden Sie Protokolle zum Abschluss von Aufträgen, um Details zu Aufträgen aufzuzeichnen, wenn diese abgeschlossen sind, ohne dass zusätzliche Kosten anfallen. Weitere Informationen finden Sie unter Die Auftragsabwicklung meldet sich an AWS 5 STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
18. Juni 2025	AWS PCS-Veröffentlichung in AWS GovCloud (US)	AWS PCS ist jetzt verfügbar in AWS GovCloud (US-East) (us-gov-east-1) und AWS GovCloud (US-West() (US-Regierung West-1). CloudFormation Vorlagen für den Einstieg finden Sie im AWS GovCloud (US) Regions. Weitere Informationen erhalten Sie unter Verwenden Sie CloudFormation um eine Probe zu erstellen AWS PCS-Cluster und CloudFormation Vorlagen zum Erstellen einer Probe AWS PCS-Cluster. Für weitere Informationen über die AWS PCS-Serviceendpunkte in AWS GovCloud (US) Regions, siehe Endpunkte und Servicekontingente für AWS 5 STÜCK. Für weitere Informationen zu den Unterschieden in AWS GovCloud (US) Regions, siehe AWS PCS rein AWS GovCloud (US)	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
		in der AWS GovCloud (US) Benutzerleitfaden .	
18. Juni 2025	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.2.1-1. Weitere Informationen finden Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK.	Nicht zutreffend
15. Mai 2025	Neues Feature: Buchhaltung	Die Slurm-Buchhaltung wird jetzt für Slurm 24.11 oder höher unterstützt. Weitere Informationen finden Sie unter Slurm-Buchhaltung in AWS 5 STCK.	AWS SDK: 15.05.2025

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
15. Mai 2025	Für Slurm 24.11 aktualisiert	Das Benutzerhandbuch für die Unterstützung von Slurm 24.11.5 wurde aktualisiert. Weitere Informationen finden Sie hier: • Slurm-Versionen in AWS 5 STCK • Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK • Versionshinweise für AWS PCS-Beispiel-AMIs	Nicht zutreffend
5. Mai 2025	Häufig gestellte Fragen zu Slurm-Versionen aktualisiert	Die häufig gestellten Fragen (FAQ) der Slurm-Versionen zu Slurm-Versionen, die sich dem Ende des Lebenszyklus (EOL) nähern oder darüber hinaus sind, wurden aktualisiert. Weitere Informationen finden Sie unter Häufig gestellte Fragen zu Slurm-Versionen in AWS 5 STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
17. April 2025	Neues Thema: Wie erhalte ich Informationen zur Compute-Node-Gruppe	Erfahren Sie, wie Sie Details für eine erhalten AWS PCS-Rechenknotengruppe, z. B. ihre ID, ARN und AMI-ID. Weitere Informationen finden Sie unter Details zur Compute-Knotengruppe in AWS PCS abrufen.	Nicht zutreffend
2. April 2025	Das Slurm-Installationsprogramm wurde aktualisiert	Das AMI-Thema für den Slurm-Installer wurde am 24.05.7-1 aktualisiert. Weitere Informationen finden Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK.	Nicht zutreffend
28. März 2025	Kontingente für die maximale Anzahl von Rechenknotengruppen und Warteschlangen hinzugefügt	Interne, nicht anpassbare Kontingente für die maximale Anzahl von Rechenknotengruppen pro Cluster und die maximale Anzahl von Warteschlangen pro Cluster wurden hinzugefügt. Weitere Informationen finden Sie unter Interne Kontingente.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
14. März 2025	Hat einen Eigenschaftenschlüssel im geändert CloudFormation Muster	`Id` ist jetzt `TemplateId` für die `CustomLaunchTemplate` Eigentum in der CloudFormation Vorlage. Weitere Informationen finden Sie unter Ressourcen in Teile einer CloudFormation Vorlage für AWS 5 STCK.	Nicht zutreffend
13. März 202	Versionsinformationen für die hinzugefügt AWS PCS-Agent und Slurm	Es wurde ein neues Thema hinzugefügt, das die Änderungen für jede Version von beschreibt AWS PCS-Agent. Weitere Informationen finden Sie unter AWS PCS-Agent-Versionen. Zum Thema Slurm-Versionen wurden weitere Informationen hinzugefügt, die wichtige Supportdaten und detaillierte Versionshinweise für beschreiben AWS PCS-Unterstützung für Slurm. Weitere Informationen finden Sie unter Slurm-Versionen in AWS 5 STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
7. März 2025	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.2.0-1. Weitere Informationen finden Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK.	Nicht zutreffend
3. Februar 2025	Es wurde ein Thema zur Verwendung hinzugefügt AWS CloudFormation mit AWS STCK	Dem Benutzerhandbuch wurde ein Thema hinzugefügt, das ein Beispiel für die Verwendung enthält CloudFormation mit AWS STCK. Das Thema enthält ein Verfahren zur Verwendung eines Beispiels CloudFormation Vorlage zum Erstellen des Beispiels AWS PCS-Cluster und beschreibt kurz die Abschnitte dieser Vorlage. Weitere Informationen finden Sie unter Erste Schritte mit CloudFormation AWS PCS.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
18. Dezembe	Für Slurm 24.05 aktualisiert	Das Benutzerhandbuch für den Slurm 24.05-Support wurde aktualisiert. Weitere Informationen erhalten Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK und Versionshinweise für AWS PCS-Beispiel-AMIs.	Nicht zutreffend
18. Dezembe	Aktualisierte NVIDIA-Versionen für Slurm 23.11-Beispiel-AMIs	Die NVIDIA-Treiber- und CUDA-Versionen in den Slurm 23.11-Beispiel-AMIs wurden aktualisiert. Weitere Informationen finden Sie unter Versionshinweise für AWS PCS-Beispiel-AMIs.	Nicht zutreffend
17. Dezembe	Das Slurm-Installationsprogramm wurde aktualisiert	Das AMI-Thema für den Slurm-Installer 23.11.10-3 wurde aktualisiert. Weitere Informationen finden Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
13. Dezember 2024	PCS-Agent aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.1.1-1. Weitere Informationen finden Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK.	Nicht zutreffend
6. Dezember 2024	PCS-Agent und Slurm-Installationsprogramm aktualisiert	Das AMI-Thema für wurde aktualisiert AWS PCS-Agent 1.1.0-1 und Slurm-Installer 23.11.10-2. Weitere Informationen finden Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK.	Nicht zutreffend
6. Dezember 2024	Es wurde ein Thema zur Betriebssystemunterstützung hinzugefügt	Weitere Informationen finden Sie unter Unterstützte Betriebssysteme in AWS PCS.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
8. November	Reorganisiertes Benutzerhandbuch	Wir haben das Benutzerhandbuch neu organisiert, um Themen auf die oberste Ebene zu bringen, einige Themen auf eigene Seiten verschoben und ähnliche Themen zusammengefasst.	Nicht zutreffend
7. November	Die AMI-Themen wurden aktualisiert	Das AMI-Thema für Slurm 23.11.10 und libjwt 17.0 wurde aktualisiert. Weitere Informationen erhalten Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK und Schritt 3 — Slurm installieren. Die Versionshinweise für AMIs wurden vereinfacht und korrigiert. Weitere Informationen finden Sie unter Versionshinweise für AWS PCS-Beispiel-AMIs.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
7. November	Es wurde ein neues Thema zur Verwendung verschlüsselter EBS-Volumes mit hinzugefügt AWS STCK	Es wurde ein Thema hinzugefügt, das die KMS-Schlüsselrichtlinie beschreibt, die für verschlüsselte EBS-Volumes erforderlich ist AWS STCK. Weitere Informationen finden Sie unter Erforderliche KMS-Schlüsselrichtlinie für die Verwendung mit verschlüsselten EBS-Volumes in AWS STCK.	Nicht zutreffend
18. Oktober 2024	AWS PCS-Agent 1.0.1-1 veröffentlicht	Aktualisiert AMI-related Dokumentation, auf die verwiesen werden kann AWS PCS-Agent-Version 1.0.1-1. Weitere Informationen erhalten Sie unter Softwareinstallationsprogramme zum Erstellen benutzerdefinierter AMIs für AWS 5 STCK und Schritt 2 — Installieren Sie den AWS PCS-Agent.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
10. Oktober 2	Es wurde ein Kapitel zur Problembehandlung hinzugefügt	Es wurde ein Kapitel zur Problembehandlung mit einem Thema hinzugefügt, in dem es darum geht, dass EC2-Instances nach einem Neustart automatisch ersetzt werden. Weitere Informationen finden Sie unter Behebung von Problemen in AWS Dienst für parallele Datenverarbeitung.	Nicht zutreffend
23. Septembe 2024	Die Mindestberechtigungen für die Verwendung von API-Aktionen und für einen Dienstadministrator wurden aktualisiert	Das `ec2:DescribeInstanceTypeOfferings` Eine Genehmigung ist jetzt erforderlich für `CreateComputeNodeGroup` und `UpdateComputeNodeGroup` API-Aktionen. Weitere Informationen finden Sie unter Mindestberechtigungen für AWS 5 STÜCK.	Nicht zutreffend
5. Septembe 2024	Die Beispiel-IAM-Richtlinie für die Mindestberechtigungen für einen Dienstadministrator wurde aktualisiert	Weitere Informationen finden Sie unter Mindestberechtigungen für einen Dienstadministrator.	Nicht zutreffend

Date	Änderungen	Aktualisierungen der Dokumentation	API-Versionen aktualisiert
5. Septembe 2024	Dem JSON auf der Seite mit verwalteten Richtlini en wurde eine fehlende Berechtigung hinzugefügt	Dies war nur eine Korrektur der Dokumenta tion. Die tatsächlich verwaltete Richtlinie wurde nicht geändert. Weitere Informationen finden Sie unter AWS verwaltete Richtlinien für AWS Dienst für parallele Datenverarbeitung.	Nicht zutreffend
28. August 2(	Die Seite „Verwalte te Richtlinien“ wurde hinzugefügt	Weitere Informationen finden Sie unter AWS verwaltete Richtlinien für AWS Dienst für parallele Datenverarbeitung.	Nicht zutreffend
28. August 2(	AWS PCS-Version	Erste Veröffentlichung des AWS PCS-Benut zerhandbuch.	AWS SDK: 2024-08-28

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.