



Management Guide

Amazon EMR



Amazon EMR: Management Guide

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

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

Table of Contents

Was ist Amazon EMR?	1
Verstehen, wie man Amazon EMR-Cluster erstellt und mit ihnen arbeitet	1
Machen Sie sich mit Clustern und Knoten vertraut	2
Übermitteln von Aufträgen an einen Cluster	2
Verarbeiten von Daten	3
Verstehen des Cluster-Lebenszyklus	5
Vorteile der Verwendung von Amazon EMR	6
Kosteneinsparungen	7
AWS Integration	8
Bereitstellung	8
Skalierbarkeit und Flexibilität	9
Zuverlässigkeit	9
Sicherheit	10
Überwachen	12
Verwaltungsschnittstellen	12
Architektur- und Service-Ebenen	13
Speicher	13
Cluster-Ressourcenverwaltung	14
Datenverarbeitungs-Frameworks	15
Anwendungen und Programme	16
Bevor Sie Amazon EMR einrichten	17
Melde dich an für ein AWS-Konto	17
Erstellen eines Benutzers mit Administratorzugriff	17
Erstellen Sie ein EC2 Amazon-Schlüsselpaar für SSH	19
Nächste Schritte	20
Erste Schritte-Tutorial	21
Einrichtung Ihres Amazon EMR-Clusters	21
Schritt 1: Datenressourcen konfigurieren und einen Amazon EMR-Cluster starten	23
Speicher für Amazon EMR vorbereiten	23
Bereiten Sie eine Anwendung mit Eingabedaten für Amazon EMR vor	23
Starten eines Amazon-EMR-Clusters	26
Schritt 2: Arbeiten an Ihren Amazon EMR-Cluster einreichen	29
Arbeit einreichen und Ergebnisse ansehen	29
Ergebnisse anzeigen	33

Schritt 3: Bereinigen	37
So beenden Sie Ihren Cluster	37
Löschen von S3-Ressourcen	39
Nächste Schritte	39
Erkunden Sie Big-Data-Anwendungen für Amazon EMR	40
Planen Sie Cluster-Hardware, Netzwerke und Sicherheit	40
Verwalten von Clustern	40
Verwenden Sie eine andere Schnittstelle	40
Stöbern Sie im technischen Blog von EMR	40
Verwaltung von Amazon EMR-Clustern mit der Konsole	41
Funktionen der Konsole	41
Zusammenfassung der Unterschiede	42
Cluster-Kompatibilität in der Konsole	42
Cluster erstellen	43
Cluster anzeigen und nach ihnen suchen	44
Cluster-Details anzeigen oder bearbeiten	45
Unterschiede bei der Arbeit mit Sicherheitskonfigurationen	46
Amazon EMR Studio	48
Schlüsselfeatures	48
Feature-Verlauf	49
Funktionsweise	50
Authentifizierung und Benutzeranmeldung	51
Zugriffskontrolle	55
Workspaces	56
Notebook-Speicher	57
Funktionen, Anforderungen und Einschränkungen von EMR Studio	58
Überlegungen	58
Bekannte Probleme	60
Feature-Einschränkungen	62
Service Limits	63
Bewährte Methoden für VPC und Subnetze für EMR Studio	63
Cluster-Voraussetzungen	64
EMR Studio konfigurieren	66
Administratorberechtigungen zum Erstellen eines EMR-Studios	67
Richten Sie ein EMR Studio ein	73
Amazon EMR Studio-Ressourcen überwachen, aktualisieren und löschen	142

Workspace-Notizbücher verschlüsseln	149
Den EMR-Studio-Netzwerkverkehr steuern	152
Cluster-Vorlagen erstellen	155
Zugriff und Berechtigungen für Git-basierte Repositorien	161
Spark-Aufträge optimieren	165
Verwenden Sie EMR Studio	166
Lernen Sie EMR Studio-Arbeitsbereiche kennen	167
Zusammenarbeit im Workspace	175
Einen Workspace mit einer Laufzeit-Rolle ausführen	179
Führen Sie Amazon EMR Studio Workspace Workspace-Notebooks programmgesteuert aus	184
Durchsuchen Sie Daten mit SQL Explorer für EMR Studio	185
Ordnen Sie einen Computer einem Workspace zu	186
Git-Repositorys verknüpfen	194
Athena-Integration	198
CodeWhisperer Integration	199
Debuggen Sie Anwendungen und Aufträge	201
Installieren Sie Kernel und Bibliotheken	206
Magische Befehle	207
Verwenden Sie mehrsprachige Notebooks mit Spark-Kernen	217
EMR Notebooks	220
Notizbücher in der Konsole	221
Über den Übergang	221
Was müssen Sie als Nächstes tun?	222
Vorteile von Workspace	222
Erforderliche Berechtigungen	223
Überlegungen	224
Cluster-Voraussetzungen	225
Unterschiede in den Funktionalitäten nach Cluster-Release-Version	226
Limits für gleichzeitig angefügte EMR-Notebooks	227
Jupyter Notebook und Python-Versionen	228
Sicherheitsüberlegungen	228
Erstellen eines Notebooks	228
Arbeiten mit EMR-Notebooks	232
Grundlegendes zum Notebook-Status	233
Arbeiten mit dem Notebook-Editor	234

Wechseln von Clustern	236
Löschen von Notebooks und Notebook-Dateien	236
Freigeben von Notebook-Dateien	237
Beispiele für programmatische Befehle für EMR Notebooks	238
Übersicht	238
Berechtigungen	239
Einschränkungen	241
Beispiele	241
Beispiele für Notebook-CLI-Befehle	241
Boto3-SDK-Beispielskript	247
Ruby-Beispielskript	250
Benutzer-Identitätswechsel für Spark	252
Einrichten der Spark-Benutzererkennung	253
Verwenden des Spark-Widgets für die Auftragsüberwachung	254
Sicherheit	255
Installation und Verwendung von Kernen und Bibliotheken in EMR Studio	256
.....	257
Installieren von Kernen und Python-Bibliotheken auf einem Cluster-Primärknoten	257
Überlegungen und Einschränkungen bei Bibliotheken für Notebooks	260
Arbeiten mit Notebook-Bibliotheken	260
Verknüpfen von Git-basierten Repositorys mit EMR Notebooks	262
Voraussetzungen und Überlegungen	263
Hinzufügen eines Git-basierten Repositorys zu Amazon EMR	266
Ein Git-basiertes Repository aus einem EMR Studio Workspace aktualisieren oder löschen	267
Ein Git-basiertes Repository in EMR Studio verknüpfen oder die Verknüpfung aufheben	268
Erstellen Sie ein neues Notebook mit einem zugehörigen Git-Repository in EMR Studio	270
Verwenden Sie Git-Repositorys in einem EMR Studio-Notebook	271
Planung, Konfiguration und Start von Amazon EMR-Clustern	272
Schnelles Starten eines Amazon EMR-Clusters	272
Amazon EMR-Clusterstandort und Datenspeicher konfigurieren	273
Wählen Sie eine AWS Region für Ihren Amazon EMR-Cluster	274
Arbeiten mit Speicher- und Dateisystemen	275
Eingabedaten für die Verarbeitung mit Amazon EMR vorbereiten	280
Einen Speicherort für die Amazon EMR-Clusterausgabe konfigurieren	300
Planung und Konfiguration primärer Knoten in Ihrem Amazon EMR-Cluster	307

Funktionen, die Hochverfügbarkeit in einem Amazon EMR-Cluster unterstützen, und wie sie mit Open-Source-Anwendungen funktionieren	308
Amazon-EMR-Clustern mit mehreren Primärknoten starten	319
Amazon EMR-Integration mit EC2 Platzierungsgruppen	325
Überlegungen und bewährte Methoden für einen Cluster mit mehreren Primärknoten	333
EMR-Cluster auf AWS Outposts	335
Voraussetzungen	336
Einschränkungen	336
Überlegungen zur Netzwerkkonnektivität	337
Erstellen eines Amazon EMR-Clusters auf AWS Outposts	338
EMR-Cluster in AWS Local Zones	339
Unterstützte Instance-Typen	339
Erstellen eines Amazon-EMR-Clusters auf Local Zones	340
Docker für die Verwendung mit Amazon EMR-Clustern konfigurieren	341
Docker-Registrierungen	342
Konfigurieren von Docker-Registrierungen	343
YARN für den Zugriff auf Amazon ECR auf EMR 6.0.0 und früher konfigurieren	344
Steuern Sie die Kündigung des Amazon EMR-Clusters	346
Konfiguration eines Amazon EMR-Clusters zum Fortfahren oder Beenden nach der Ausführung des Schritts	347
Verwenden einer Richtlinie zur automatischen Beendigung	350
Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen	357
Ersetzen fehlerhafter Knoten durch Amazon EMR	363
Standardeinstellungen für den Austausch von Knoten und den Kündigungsschutz	365
Konfiguration des Austauschs fehlerhafter Knoten beim Start eines Clusters	365
Konfiguration eines fehlerhaften Knotenaustauschs in einem laufenden Cluster	367
Arbeiten mit AMIs	368
Übersicht	368
Verwenden des Standard-AMI	369
Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration	542
Ändern Sie die AL-Version	556
Anpassen des EBS-Root-Volumes	557
Konfigurieren Sie Anwendungen, wenn Sie Ihren Amazon EMR-Cluster starten	560
Erstellen Sie Bootstrap-Aktionen	561

Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren	567
Grundsätzliches zu Knotentypen	568
EC2 Amazon-Instance-Typen für die Verwendung mit Amazon EMR konfigurieren	571
Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren	1561
Standardmäßige Protokolldateien	1561
Archivieren von Protokolldateien in Amazon S3	1562
Protokollspeicherorte	1569
Taggen und kategorisieren Sie Amazon EMR-Cluster-Ressourcen	1570
Einschränkungen, die für das Markieren von Ressourcen in Amazon EMR gelten	1572
Amazon EMR-Ressourcen für die Abrechnung kennzeichnen	1572
Hinzufügen von Tags zu einem Amazon EMR-Cluster	1573
Tags auf einem Amazon EMR-Cluster anzeigen	1575
Tags aus einem Amazon EMR-Cluster entfernen	1576
Integration von Treibern und Drittanbieteranwendungen auf Amazon EMR	1577
Verwenden von Business-Intelligence-Tools in Amazon EMR	1577
Sicherheit	1578
Netzwerk- und Infrastruktursicherheit	1578
Updates des standardmäßigen Amazon Linux AMI	1579
AWS Identity and Access Management mit Amazon EMR	1580
Cluster mit einem Mandanten und mehreren Mandanten	1581
Datenschutz	1582
Datenzugriffskontrolle	1582
Sicherheitskonfigurationen	1583
Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI	1584
Eine Sicherheitskonfiguration angeben	1616
Datenschutz	1617
Verschlüsseln Sie Daten im Ruhezustand und bei der Übertragung mit Amazon EMR	1618
IAM mit Amazon EMR	1659
Zielgruppe	1659
Authentifizierung mit Identitäten	1660
Verwalten des Zugriffs mit Richtlinien	1664
Funktionsweise von Amazon EMR mit IAM	1667
Schritte für Laufzeit-Rollen für Amazon EMR	1675
Konfigurieren von Servicerollen für Amazon EMR	1684
Beispiele für identitätsbasierte Richtlinien	1748

S3 Access Grants mit Amazon EMR	1790
Übersicht	1790
Funktionsweise	1791
Überlegungen	1792
Starten Sie einen Cluster.	1793
Lake Formation	1794
fallbackToIAM	1795
Authentifizieren von Cluster-Knoten	1796
Verwenden Sie ein EC2 key pair für SSH-Anmeldeinformationen für Amazon EMR	1796
Verwendung der Kerberos-Authentifizierung	1797
Verwendung der LDAP-Authentifizierung	1837
Amazon EMR in Identity Center integrieren	1849
Übersicht	1850
Features	1850
Erste Schritte	1851
Überlegungen	1859
Integrieren von Amazon EMR mit Lake Formation	1861
Wie Amazon EMR mit Lake Formation funktioniert	1861
Voraussetzungen	1862
Wie Amazon EMR mit Lake Formation funktioniert	1863
Hudi und Lake Formation	1868
Iceberg und Lake Formation	1870
Delta Lake und Lake Formation	1871
Überlegungen	1873
Verwenden einer Lake Formation Formation-Job-Laufzeitrolle mit vollständigem Tabellenzugriff	1874
Integrieren Sie Amazon EMR mit Apache Ranger	1882
Übersicht über Ranger	1883
Überlegungen zur Verwendung von Amazon EMR mit Apache Ranger	1885
Amazon EMR für Apache Ranger einrichten	1888
Apache Ranger-Plugins für Amazon EMR-Integrationsszenarien	1906
Fehlerbehebung für Apache Ranger	1934
Arbeiten mit AWS Glue-Datenkatalogansichten in Amazon EMR (Vorschau)	1938
Erstellen einer Data-Catalog-Ansicht	1939
Zugriff auf eine Datenkatalogansicht aktivieren	1941
Abfrage einer Data-Catalog-Ansicht	1943

Einschränkungen	1943
Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster	1944
Arbeiten mit von Amazon EMR verwalteten Sicherheitsgruppen	1946
Arbeiten mit zusätzlichen Sicherheitsgruppen für einen Amazon EMR-Cluster	1957
Angaben von Sicherheitsgruppen	1958
Sicherheitsgruppe für EMR Notebooks	1961
Blockieren des öffentlichen Zugriffs	1963
Compliance-Validierung	1969
Ausfallsicherheit	1970
Sicherheit der Infrastruktur	1970
Herstellen einer Verbindung mit Amazon EMR über einen Schnittstellen-VPC-Endpunkt ...	1971
Amazon EMR-Cluster verwalten	1976
Eine Verbindung zu einem Amazon-EMR-Cluster herstellen	1976
Bevor Sie sich verbinden	1977
Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her	1979
Arbeit an einen Amazon EMR-Cluster einreichen	2006
Schritte mit der Konsole hinzufügen	2008
Schritte mit der CLI hinzufügen	2011
Ausführen mehrerer Schritte	2013
Schritte nach dem Einreichen von Arbeiten an einen Amazon EMR-Cluster anzeigen	2014
Schritte abbrechen, wenn Sie Arbeit an einen Amazon EMR-Cluster einreichen	2015
Einen Amazon EMR-Cluster bei der Ausführung seiner Arbeit anzeigen und überwachen	2016
Status und Details des Amazon EMR-Clusters anzeigen	2017
Verbessertes schrittweises Debugging mit Amazon EMR	2022
Amazon EMR-Anwendungsverlauf anzeigen	2024
Amazon EMR-Protokolldateien anzeigen	2037
Cluster-Instances in Amazon anzeigen EC2	2041
CloudWatch Ereignisse und Metriken von Amazon EMR	2043
Cluster-Anwendungsmetriken mithilfe von Ganglia mit Amazon EMR anzeigen	2136
CloudTrail protokolliert	2136
Bewährte Methoden zur EMR-Observabilität	2141
Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen	2142
Überlegungen	2143
Verwaltete Skalierung	2144

Auto Scaling mit einer benutzerdefinierten Richtlinie	2190
Die Größe eines aktiven Clusters anpassen	2204
Timeouts bei der Bereitstellung	2212
Cluster-Scale-Down-Optionen für Amazon EMR-Cluster	2216
Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“	2221
Von der Konsole aus beenden	2222
Über CLI beenden	2222
Über API beenden	2223
Einen Cluster klonen	2224
Automatisieren Sie wiederkehrende Amazon EMR-Cluster mit AWS Data Pipeline	2225
Amazon EMR-Tutorials	2226
Amazon EMR on EC2 — Verbesserte Überwachung mit CloudWatch benutzerdefinierten Metriken und Protokollen	2226
Übersicht	2226
Voraussetzungen und Hintergrund	2226
.....	2227
Problembehandlung bei Amazon EMR-Clustern	2241
Tools zur Fehlerbehebung	2241
Anzeigen von Cluster-Details	2242
Anzeigen von Fehlerdetails	2242
Führen Sie Skripts aus und konfigurieren Sie Prozesse	2243
Anzeige von -Protokolldateien	2243
Überwachen Sie die Leistung des Clusters	2244
Prozesse anzeigen und neu starten	2244
Anzeigen von ausgeführten Prozessen	2245
Beenden und Neustarten von Prozessen	2247
Häufige Fehler	2249
Fehlercodes	2250
Ressourcenfehler bei Amazon EMR-Clustervorgängen	2265
Cluster-Eingabe- und Ausgabefehler bei Amazon EMR-Vorgängen	2280
Berechtigungsfehler bei Amazon EMR-Clustervorgängen	2283
Hive-Cluster-Fehler	2284
VPC-Fehler bei Amazon EMR-Clusteroperationen	2286
Fehler beim Streaming von Amazon EMR-Clustern	2290
Amazon EMR: Fehler beim benutzerdefinierten JAR-Cluster	2292
Fehler bei Amazon EMR AWS GovCloud (US-West)	2292

Finden Sie einen fehlenden Cluster	2293
Fehlerbehebung für ausgefallene Cluster	2293
Schritt 1: Sammeln Sie Daten über das Problem mit dem Amazon EMR-Cluster	2294
Schritt 2: Die Umgebung prüfen	2295
Schritt 3: Die letzte Statusänderung überprüfen	2296
Schritt 4: Untersuchen Sie die Amazon EMR-Protokolldateien	2297
Schritt 5: Testen Sie den Amazon EMR-Cluster Schritt für Schritt	2298
Fehlerbehebung für einen langsamen Cluster	2299
Schritt 1: Sammeln Sie Daten über das Problem mit dem Amazon EMR-Cluster	2300
Schritt 2: Überprüfen Sie die EMR-Cluster-Umgebung	2301
Schritt 3: Untersuchen Sie die Protokolldateien für den Amazon EMR-Cluster	2303
Schritt 4: Überprüfen Sie den Zustand des Amazon EMR-Clusters und der Instance	2305
Schritt 5: Nach gesperrten Gruppen suchen	2306
Schritt 6: Überprüfen Sie die Konfigurationseinstellungen für den Amazon EMR-Cluster	2307
Schritt 7: Untersuchen Sie die Eingabedaten für den Amazon EMR-Cluster	2310
Behebung häufiger Probleme bei der Verwendung von Amazon EMR mit AWS Lake Formation	2311
Der Zugriff auf den Data Lake ist nicht zulässig	2311
Sitzungsablauf	2311
Keine Berechtigungen für Benutzer in der angeforderten Tabelle	2311
Abfragen von kontenübergreifenden Daten, die mit Lake Formation geteilt wurden	2312
Einfügen in, Erstellen und Ändern von Tabellen	2313
Schreiben Sie Anwendungen, die Amazon EMR-Cluster starten und verwalten	2314
End-to-end Beispiel für Amazon EMR Java-Quellcode	2314
Allgemeine Konzepte für Amazon EMR-API-Aufrufe	2319
Endpunkte für Amazon EMR	2319
Angaben von Cluster-Parametern in Amazon EMR	2319
Availability Zones in Amazon EMR	2320
So verwenden Sie weitere Dateien und Bibliotheken in Amazon-EMR-Clustern	2321
Verwenden Sie SDKs , um Amazon EMR anzurufen APIs	2321
Verwenden von AWS SDK für Java , um einen Amazon EMR-Cluster zu erstellen	2321
Amazon EMR Service Quotas verwalten	2324
Was sind Amazon EMR Service Quotas?	2324
Amazon EMR Service Quotas verwalten	2325
Wann sollten EMR-Ereignisse eingerichtet werden in CloudWatch	2326
AWS Glossar	2330

Dokumentverlauf 2331

..... mmcccxxxii

Was ist Amazon EMR?

Amazon EMR, zuvor Amazon Elastic genannt MapReduce, ist eine verwaltete Cluster-Plattform, die die Ausführung von Big-Data-Frameworks wie [Apache Hadoop](#) und [Apache Spark](#) vereinfacht, AWS um riesige Datenmengen zu verarbeiten und zu analysieren. Die Verwendung dieser Frameworks und verwandter Open-Source-Projekte, können Sie Daten zu Analysezielen und Business-Intelligence-Workloads verarbeiten. Amazon EMR zum Transformieren und Verschieben lässt auch große Datenmengen in und aus anderen AWS -Datenspeichern und Datenbanken verwenden, wie z. B. Amazon Simple Storage Service (Amazon S3) und Amazon DynamoDB.

Wenn Sie Amazon EMR erstmalig verwenden, empfehlen wir, zusätzlich zu diesem Abschnitt die folgenden Abschnitte zu lesen:

- [Amazon EMR](#) – Auf dieser Service-Seite finden Sie die Highlights, Produktdetails und Preisinformationen.
- [Tutorial: Erste Schritte mit Amazon EMR](#) – Mit diesem Tutorial können Sie schnell mit Amazon EMR beginnen.

In diesem Abschnitt

- [Verstehen, wie man Amazon EMR-Cluster erstellt und mit ihnen arbeitet](#)
- [Vorteile der Verwendung von Amazon EMR](#)
- [Amazon EMR-Architektur und Service-Ebenen](#)

Verstehen, wie man Amazon EMR-Cluster erstellt und mit ihnen arbeitet

Dieses Thema bietet eine Übersicht über die Amazon-EMR-Cluster, einschließlich der Übermittlung von Aufträgen an einen Cluster, der Verarbeitung von Daten und der verschiedenen Status, die der Cluster während der Verarbeitung durchläuft.

In diesem Thema

- [Machen Sie sich mit Clustern und Knoten vertraut](#)
- [Übermitteln von Aufträgen an einen Cluster](#)
- [Verarbeiten von Daten](#)

- [Verstehen des Cluster-Lebenszyklus](#)

Machen Sie sich mit Clustern und Knoten vertraut

Die zentrale Komponente des Amazon EMR ist der Cluster. Ein Cluster ist eine Sammlung von Amazon Elastic Compute Cloud (Amazon EC2) -Instances. Jede Instance in einem Cluster wird als Knoten bezeichnet. Jeder Knoten verfügt über eine Rolle im Cluster – Knotentyp genannt. Amazon EMR installiert auch verschiedene Softwarekomponenten auf den einzelnen Knotentypen und überträgt so jedem Knoten eine Rolle in einer verteilten Anwendung wie Apache Hadoop.

Amazon EMR verfügt über die folgenden Knotentypen:

- **Primärknoten:** Knoten, der den Cluster durch die Ausführung von Softwarekomponenten verwaltet, die die Verteilung von Daten und Aufgaben auf andere Knoten zur Verarbeitung koordinieren. Der Primärknoten überwacht den Status der Aufgaben und überwacht den Zustand des Clusters. Jeder Cluster verfügt über einen Primärknoten und es ist möglich, einen Einzelknoten-Cluster nur mit dem Primärknoten zu erstellen.
- **Core-Knoten:** Knoten mit Software-Komponenten, die Aufgaben ausführen und Daten im HDFS (Hadoop Distributed File System) auf dem Cluster speichern. Multiknoten-Cluster enthalten mindestens einen Core-Knoten.
- **Aufgabenknoten:** Knoten mit Software-Komponenten, die nur Aufgaben ausführen und keine Daten in HDFS speichern. Aufgabenknoten sind optional.

Übermitteln von Aufträgen an einen Cluster

Bei Ausführung eines Clusters in Amazon EMR haben Sie mehrere Möglichkeiten, die auszuführende Arbeit anzugeben.

- Stellen Sie die gesamte Definition der auszuführenden Arbeit in Funktionen bereit, die Sie als Schritte angeben, wenn Sie einen Cluster erstellen. Dies wird in der Regel für Cluster durchgeführt, die eine bestimmte Datenmenge verarbeiten und nach Abschluss der Verarbeitung beendet werden.
- Erstellen Sie einen Cluster mit langer Laufzeit und verwenden Sie die Amazon EMR-Konsole, die Amazon EMR-API oder die AWS CLI zum Senden von Schritten, die einen oder mehrere Jobs enthalten können. Weitere Informationen finden Sie unter [Arbeit an einen Amazon EMR-Cluster einreichen](#).

- Erstellen Sie einen Cluster, stellen Sie nach Bedarf eine Verbindung zum Primärknoten und zu anderen Knoten mit SSH her und verwenden Sie die von den installierten Anwendungen bereitgestellten Schnittstellen, um Aufgaben auszuführen und Abfragen zu senden entweder in Skripts oder interaktiv. Weitere Informationen finden Sie im [Handbuch zu Amazon-EMR-Versionen](#).

Verarbeiten von Daten

Wenn Sie einen Cluster starten, bestimmen Sie die zu installierenden Frameworks und Anwendungen, damit Ihren Anforderungen an die Datenverarbeitung entsprochen wird. Um Daten in Ihrem Amazon-EMR-Cluster zu verarbeiten, können Sie Aufträge oder Abfragen direkt an installierte Anwendungen senden oder alternativ die Schritte im Cluster ausführen.

Übermitteln von Aufträgen direkt an die Anwendungen

Sie können Aufträge direkt an die Software übermitteln, die auf Ihrem Amazon-EMR-Cluster installiert ist, und anschließend damit interagieren. Dazu stellen Sie in der Regel eine sichere Verbindung mit dem Primärknoten her und greifen auf die Schnittstellen und Tools zu, die für die Software, die direkt auf Ihrem Cluster ausgeführt wird, verfügbar sind. Weitere Informationen finden Sie unter [Eine Verbindung zu einem Amazon-EMR-Cluster herstellen](#).

Ausführen von Schritten zur Verarbeitung von Daten

Sie können einem Amazon-EMR-Cluster einen oder mehrere angeordnete Schritte übermitteln. Jeder Schritt ist eine Arbeitseinheit mit Anweisungen zur Verarbeitung von Daten durch auf dem Cluster installierte Software.

Es folgt ein Beispiel für einen Prozess mit vier Schritten:

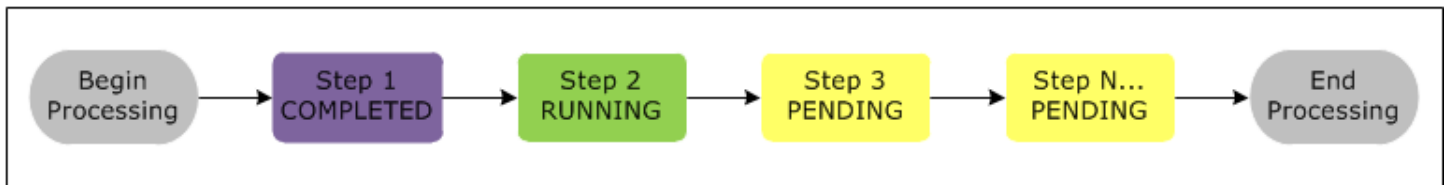
1. Übermitteln Sie die Eingabedatenmenge für die Verarbeitung.
2. Verarbeiten Sie die Ausgabe des ersten Schritts mithilfe eines Pig-Programms.
3. Verarbeiten Sie eine zweite Eingabedatenmenge mithilfe eines Hive-Programms.
4. Schreiben Sie einen Ausgabedatensatz.

Wenn Sie Daten in Amazon EMR verarbeiten, wird die Eingabe als Daten in Dateien gespeichert, die sich im zugrunde liegenden Dateisystem, wie z. B. Amazon S3 oder HDFS, befinden. Diese Daten werden während des Bearbeitungsablaufs von einem Schritt zum nächsten weitergeleitet. Im letzten Schritt werden die Ausgabedaten in einen bestimmten Speicherort geschrieben, wie zum Beispiel in einen Amazon-S3-Bucket.

Die Schritte werden in der folgenden Reihenfolge ausgeführt:

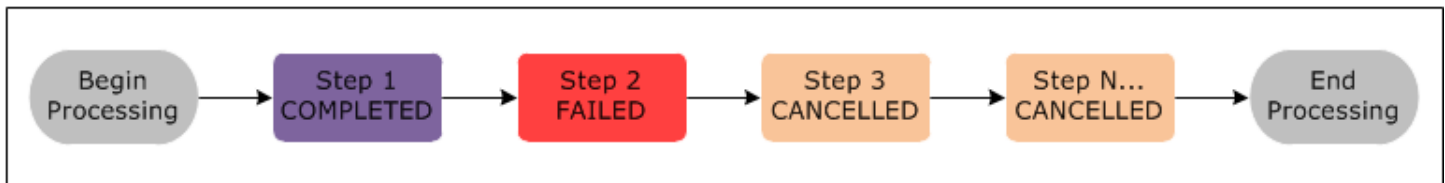
1. Eine Anfrage wird übermittelt, um mit den Verarbeitungsschritten zu beginnen.
2. Der Status aller Schritte wird auf PENDING (AUSSTEHEND) festgelegt.
3. Wenn der erste Schritt der Sequenz gestartet wird, wird dessen Status in RUNNING (WIRD AUSGEFÜHRT) geändert. Die anderen Schritte bleiben im Status PENDING (AUSSTEHEND).
4. Nachdem der erste Schritt abgeschlossen ist, wird dessen Status in COMPLETED (ABGESCHLOSSEN) geändert.
5. Der nächste Schritt der Sequenz wird gestartet und dessen Status wird in RUNNING (WIRD AUSGEFÜHRT) geändert. Nachdem er abgeschlossen ist, wird dessen Status in COMPLETED (ABGESCHLOSSEN) geändert.
6. Dieses Muster wiederholt sich für jeden Schritt, bis alle Schritte abgeschlossen sind und die Verarbeitung beendet wird.

Das folgende Diagramm stellt die Schrittsequenz sowie die Statusänderung für die einzelnen Schritte während der Verarbeitung dar.



Wenn ein Schritt während der Verarbeitung fehlschlägt, wechselt der Status zu FEHLGESCHLAGEN. Sie können für jeden Schritt festlegen, was als Nächstes geschieht. Standardmäßig werden alle verbleibenden Schritte in der Sequenz auf ABGEBROCHEN festgelegt und wenn ein vorangehender Schritt fehlschlägt. Außerdem können Sie das Ignorieren des Fehlers aktivieren, damit die verbleibenden Schritte ausgeführt werden oder der Cluster sofort beendet wird.

Das folgende Diagramm stellt die Schrittsequenz sowie die standardmäßige Statusänderung dar, wenn ein Schritt während der Verarbeitung fehlschlägt.



Verstehen des Cluster-Lebenszyklus

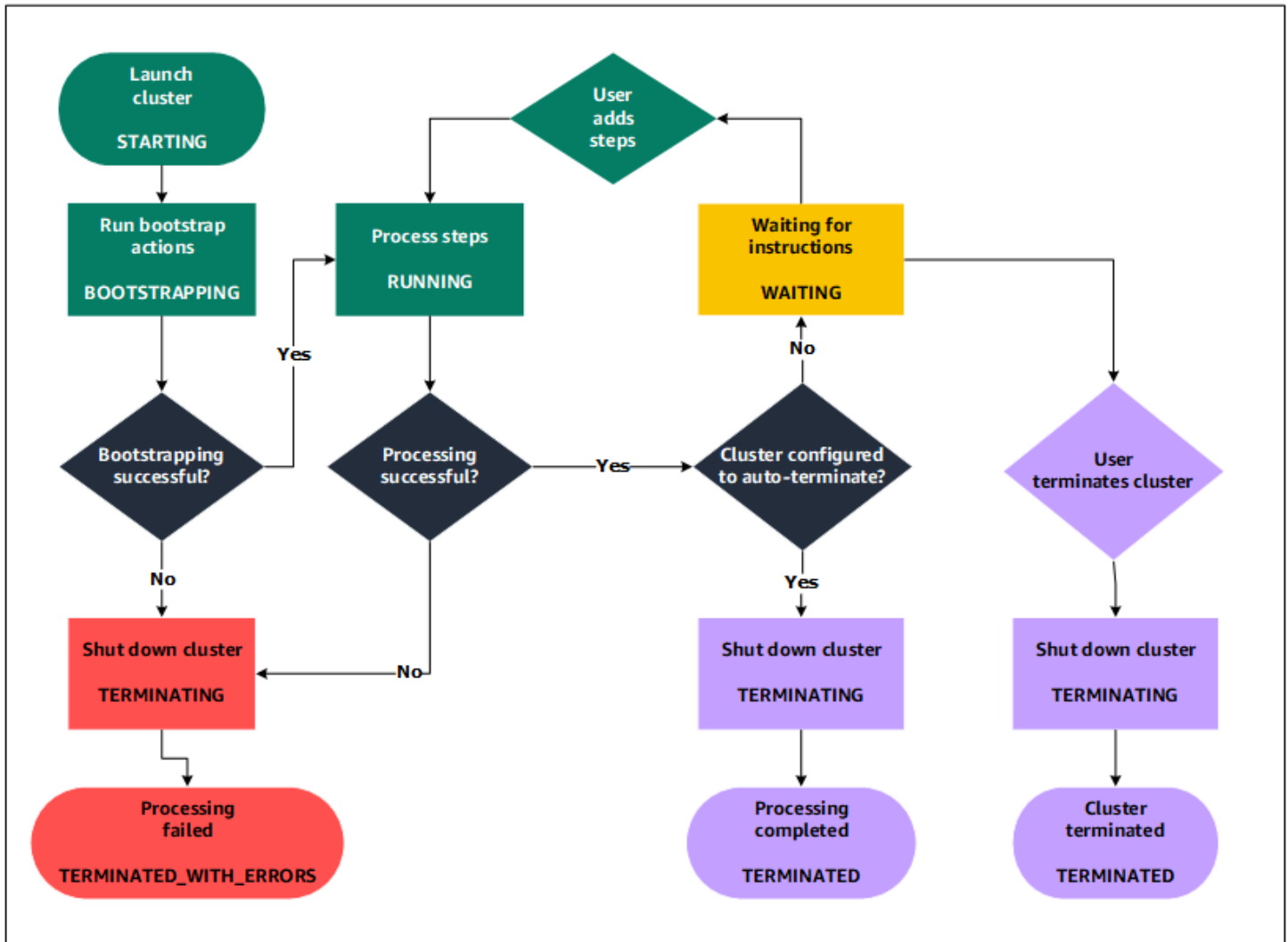
Ein erfolgreicher Amazon-EMR-Cluster befolgt diesen Prozess:

1. Amazon EMR stellt zunächst EC2 Instances im Cluster für jede Instance gemäß Ihren Spezifikationen bereit. Weitere Informationen finden Sie unter [Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren](#). Amazon EMR verwendet für alle Instances das Standard-AMI für Amazon EMR oder ein von Ihnen angegebenes benutzerdefiniertes Amazon-Linux-AMI. Weitere Informationen finden Sie unter [Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration](#). Während dieser Phase ist der Cluster-Status auf STARTING gesetzt.
2. Amazon EMR; führt Bootstrap-Aktionen aus, die Sie für jede Instance angeben. Sie können Bootstrap-Aktionen verwenden, um benutzerdefinierte Anwendungen zu installieren und erforderliche Anpassungen vorzunehmen. Weitere Informationen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#). Während dieser Phase ist der Cluster-Status auf BOOTSTRAPPING gesetzt.
3. Amazon EMR installiert die nativen Anwendungen, die Sie angeben, wenn Sie den Cluster erstellen, z. B. Hive, Hadoop, Spark usw.
4. Nachdem Bootstrap-Aktionen erfolgreich abgeschlossen und native Anwendungen installiert wurden, lautet der Cluster-Status RUNNING. An diesem Punkt können Sie die Verbindung zu Cluster-Instances herstellen. Der Cluster führt sequenziell die Schritte aus, die Sie beim Erstellen des Clusters angegeben haben. Sie können zusätzliche Schritte senden, die dann nach Abschluss der vorherigen Schritte ausgeführt werden. Weitere Informationen finden Sie unter [Arbeit an einen Amazon EMR-Cluster einreichen](#).
5. Nachdem die Schritte erfolgreich ausgeführt wurden, erhält der Cluster den Status WAITING. Wenn ein Cluster für die automatische Beendigung nach Abschluss des letzten Schritts konfiguriert ist, wechselt der Cluster den Status TERMINATING-Zustand und dann in den TERMINATED-Zustand. Wenn der Cluster so konfiguriert ist, dass er wartet, müssen Sie ihn manuell herunterfahren, wenn Sie ihn nicht mehr benötigen. Nachdem Sie den Cluster manuell beenden haben, wird dieser in den Status TERMINATING versetzt und danach in den Status TERMINATED.

Ein Fehler im Cluster-Lebenszyklus veranlasst, Amazon EMR den Cluster und dessen Instances zu beenden, sofern Sie nicht den Beendigungsschutz aktivieren. Wenn ein Cluster wegen eines Fehlers beendet wird, werden alle auf dem Cluster befindlichen Daten gelöscht und dem Cluster-Status wird der Status TERMINATED_WITH_ERRORS zugewiesen. Wenn Sie den Beendigungsschutz

aktiviert haben, können Sie Daten vom Cluster abrufen und anschließend den Beendigungsschutz entfernen und den Cluster beenden. Weitere Informationen finden Sie unter [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).

Das folgende Diagramm stellt den Lebenszyklus eines Clusters dar und wie die einzelnen Lebenszyklusphasen einem bestimmten Cluster-Status zugeordnet sind.



Vorteile der Verwendung von Amazon EMR

Es gibt zahlreiche Vorteile für die Verwendung von Amazon EMR. Dazu gehören die Flexibilität AWS und die Kosteneinsparungen, die sich im Vergleich zum Aufbau eigener Ressourcen vor Ort bieten. Dieser Abschnitt bietet eine Übersicht über die Vorteile und stellt Ihnen Links zu weiteren Informationen zur Verfügung.

Themen

- [Kosteneinsparungen](#)
- [AWS Integration](#)
- [Bereitstellung](#)
- [Skalierbarkeit und Flexibilität](#)
- [Zuverlässigkeit](#)
- [Sicherheit](#)
- [Überwachen](#)
- [Verwaltungsschnittstellen](#)

Kosteneinsparungen

Die Preise für Amazon EMR hängen vom Instance-Typ und der Anzahl der EC2 Amazon-Instances ab, die Sie bereitstellen, sowie von der Region, in der Sie Ihren Cluster starten. On-Demand-Preise bieten einen niedrigen Stundensatz, allerdings können Sie die Kosten weiter senken, indem Sie Reserved Instances erwerben oder auf Spot-Instances bieten. Spot Instances können bedeutende Kostenersparnisse bieten – in einigen Fällen betragen sie nur ein Zehntel der On-Demand-Preise.

Note

Wenn Sie Amazon S3, Amazon Kinesis oder DynamoDB mit Ihrem EMR-Cluster verwenden, fallen für diese Services zusätzliche Gebühren an, die getrennt von Ihrer Amazon-EMR-Nutzung berechnet werden.

Note

Wenn Sie einen Amazon-EMR-Cluster in einem privaten Subnetz einrichten, empfehlen wir, dass Sie auch [VPC-Endpunkte für Amazon S3](#) einrichten. Wenn sich Ihr EMR-Cluster in einem privaten Subnetz ohne VPC-Endpunkte für Amazon S3 befindet, fallen zusätzliche NAT-Gateway-Gebühren an, die mit S3-Verkehr verbunden sind, da der Verkehr zwischen Ihrem EMR-Cluster und S3 nicht innerhalb Ihrer VPC verbleibt.

Weitere Informationen zu Preisoptionen und Details finden Sie unter [Amazon-EMR-Preise](#).

AWS Integration

Amazon EMR lässt sich in andere AWS Services integrieren, um Funktionen und Funktionen in Bezug auf Netzwerk, Speicher, Sicherheit usw. für Ihren Cluster bereitzustellen. In der folgenden Liste finden Sie einige Beispiele für diese Integration:

- Amazon EC2 für die Instances, aus denen die Knoten im Cluster bestehen
- Amazon Virtual Private Cloud (Amazon VPC) zur Konfiguration des virtuellen Netzwerks, in dem Sie Ihre Instances starten
- Amazon S3 zum Speichern von Ein- und Ausgabedaten
- Amazon überwacht CloudWatch die Cluster-Leistung und konfiguriert Alarme
- AWS Identity and Access Management (IAM) zur Konfiguration von Berechtigungen
- AWS CloudTrail um Anfragen an den Service zu prüfen
- AWS Data Pipeline um Ihre Cluster zu planen und zu starten
- AWS Lake Formation um Daten in einem Amazon S3 S3-Data Lake zu entdecken, zu katalogisieren und zu sichern

Bereitstellung

Ihr EMR-Cluster besteht aus EC2 Instances, die die Arbeit ausführen, die Sie an Ihren Cluster einreichen. Wenn Sie einen Cluster starten, konfiguriert Amazon EMR die Instances mit den von Ihnen ausgewählten Anwendungen, wie beispielsweise Apache Hadoop oder Spark. Wählen Sie die Größe und den Typ der Instance aus, die am ehesten den Verarbeitungsanforderungen Ihres Clusters entsprechen: Stapelverarbeitung, schnelle Abfragen, Streaming-Daten oder große Datenspeicher. Weitere Informationen zu den für Amazon EMR verfügbaren Instance-Typen finden Sie unter [Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren](#).

Amazon EMR bietet verschiedene Möglichkeiten zum Konfigurieren von Software auf Ihrem Cluster. Sie können beispielsweise eine Amazon-EMR-Version installieren, die eine Reihe ausgewählter Anwendungen umfasst, einschließlich vielseitiger Frameworks wie Hadoop und Anwendungen, wie beispielsweise Hive, Pig oder Spark. Darüber hinaus können Sie auch eine der zahlreichen MapR-Verteilungen installieren. Amazon EMR verwendet Amazon Linux so können Sie auch Software unter Verwendung des Paket-Managers yum oder direkt von der Quelle manuell auf Ihrem Cluster installieren. Weitere Informationen finden Sie unter [Konfigurieren Sie Anwendungen, wenn Sie Ihren Amazon EMR-Cluster starten](#).

Skalierbarkeit und Flexibilität

Amazon EMR bietet Flexibilität, sodass Sie Ihren Cluster nach oben oder unten skalieren können, wenn sich Ihre Anforderungen an die Datenverarbeitung ändern. Sie können die Größe des Clusters ändern, um während Spitzenlastzeiten Instances hinzuzufügen, und um Instances zu entfernen, wenn die Spitzenlastzeiten nachlassen. So verfügen Sie über mehr Kontrolle über Ihre Kosten. Weitere Informationen finden Sie unter [Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters](#).

Amazon EMR bietet außerdem die Option, mehrere Instance-Gruppen auszuführen. So können Sie sie in einer Gruppe On-Demand-Instances verwenden, um die Verarbeitungsleistung sicherzustellen, während Sie in einer anderen Gruppe Spot Instances verwenden, um Ihre Aufträge schneller abzuschließen und Kosten zu senken. Sie können auch verschiedene Instance-Typen mischen, um die Preisvorteile von bestimmten Spot-Instance-Typen zu nutzen. Weitere Informationen finden Sie unter [Wann sollten Sie Spot Instances verwenden?](#).

Darüber hinaus bietet Amazon EMR die Flexibilität, verschiedene Dateisysteme für Ihre Eingabe-, Ausgabe- und Zwischendaten zu verwenden. Für die Verarbeitung von Daten, die Sie nicht länger als den Lebenszyklus Ihres Clusters speichern müssen, können Sie beispielsweise das Hadoop Distributed File System (HDFS) auswählen, das auf den Primär- und Core-Knoten Ihres Clusters ausgeführt wird. Sie können möglicherweise auch das EMR File System (EMRFS) für die Verwendung mit Amazon S3 auswählen. Es kann als Daten-Layer für Anwendungen auf Ihrem Cluster dienen, sodass Sie die Datenverarbeitung und den Speicher trennen und Daten außerhalb des Lebenszyklus Ihres Clusters erhalten können. EMRFS bietet Ihnen die Möglichkeit, Ihre Anforderungen an die Datenverarbeitung und an den Speicher nach oben oder nach unten zu skalieren. Sie können Ihre Anforderungen an die Datenverarbeitung skalieren, indem Sie die Größe Ihres Clusters verändern, und Ihre Speichieranforderungen skalieren, indem Sie Amazon S3 verwenden. Weitere Informationen finden Sie unter [Arbeiten mit Speicher- und Dateisystemen mit Amazon EMR](#).

Zuverlässigkeit

Amazon EMR überwacht die Knoten in Ihrem Cluster und beendet und ersetzt eine Instance automatisch, wenn ein Fehler auftritt.

Amazon EMR bietet Konfigurationsoptionen, anhand denen Sie steuern, ob der Cluster beendet werden soll automatisch oder manuell. Wenn Sie Ihren Cluster so konfigurieren, dass er automatisch beendet wird, erfolgt das, nachdem alle Schritte abgeschlossen sind. Dies wird auch als vorübergehender Cluster bezeichnet. Sie können den Cluster jedoch auch so konfigurieren, dass er nach Abschluss der Verarbeitung weiter ausgeführt wird. Auf diese Weise können Sie ihn manuell

beenden, wenn Sie ihn nicht länger benötigen. Alternativ können Sie einen Cluster erstellen, mit den installierten Anwendungen direkt interagieren und den Cluster, wenn Sie ihn nicht mehr benötigen, manuell beenden. Die Cluster in diesen Beispielen werden als langlebige Cluster bezeichnet.

Zusätzlich können Sie den Beendigungsschutz konfigurieren, um zu verhindern, dass Instances im Cluster aufgrund von Fehlern oder Problemen während der Verarbeitung beendet werden. Wenn der Beendigungsschutz aktiviert ist, können Sie die Daten vor der Beendigung von den Instances wiederherstellen. Die Standardeinstellungen für diese Optionen unterscheiden sich, je nachdem, ob Sie einen Cluster über die Konsole, die CLI oder die API starten. Weitere Informationen finden Sie unter [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).

Sicherheit

Amazon EMR nutzt andere AWS Services wie IAM und Amazon VPC sowie Funktionen wie EC2 Amazon-Schlüsselpaare, um Sie bei der Sicherung Ihrer Cluster und Daten zu unterstützen.

IAM

Amazon EMR kann mit IAM integriert werden, um Berechtigungen zu verwalten. Sie definieren Berechtigungen mit IAM-Richtlinien, die Sie Benutzern oder IAM-Gruppen anfügen. Die Berechtigungen, die Sie in den Richtlinie definieren, legen fest, welche Aktionen diese Benutzer oder Gruppenmitglieder ausführen können, und auf welche Ressourcen sie zugreifen können. Weitere Informationen finden Sie unter [Funktionsweise von Amazon EMR mit IAM](#).

Darüber hinaus verwendet Amazon EMR IAM-Rollen für den Amazon EMR-Service selbst und das EC2 Instance-Profil für die Instances. Diese Rollen gewähren dem Service und den Instances die Erlaubnis, in Ihrem Namen auf andere AWS Services zuzugreifen. Es gibt eine Standardrolle für den Amazon EMR-Service und eine Standardrolle für das EC2 Instance-Profil. Die Standardrollen verwenden AWS verwaltete Richtlinien, die automatisch für Sie erstellt werden, wenn Sie zum ersten Mal einen EMR-Cluster von der Konsole aus starten und Standardberechtigungen auswählen. Sie können die IAM-Standardrollen auch über die AWS CLI erstellen. Wenn Sie stattdessen die Berechtigungen verwalten möchten AWS, können Sie benutzerdefinierte Rollen für das Service- und Instanzprofil auswählen. Weitere Informationen finden Sie unter [Konfigurieren Sie IAM-Service rollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen](#).

Sicherheitsgruppen

Amazon EMR verwendet Sicherheitsgruppen, um den ein- und ausgehenden Datenverkehr zu Ihren Instances zu kontrollieren. EC2 Wenn Sie Ihren Cluster starten, verwendet Amazon EMR eine

Sicherheitsgruppe für Ihre primäre Instance und eine Sicherheitsgruppe, die von Ihren core/task instances. Amazon EMR configures the security group rules to ensure communication among the instances in the cluster. Optionally, you can configure additional security groups and assign them to your primary and core/task Instances gemeinsam genutzt wird, um erweiterte Regeln zu erstellen. Weitere Informationen finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Verschlüsselung

Amazon EMR unterstützt die optionale Amazon S3 serverseitige und clientseitige Verschlüsselung mit EMRFS, um die von Ihnen in Amazon S3 gespeicherten Daten zu schützen. Bei der serverseitigen Verschlüsselung werden Ihre Daten von Amazon S3 nach dem Hochladen verschlüsselt.

Bei der clientseitigen Verschlüsselung erfolgt der Ver- und Entschlüsselungsvorgang im EMRFS-Client auf Ihrem EMR-Cluster. Sie verwalten den Root-Schlüssel für die clientseitige Verschlüsselung entweder mit dem AWS Key Management Service (AWS KMS) oder Ihrem eigenen Schlüsselverwaltungssystem.

Weitere Informationen finden Sie unter [Amazon-S3-Verschlüsselung mithilfe von EMRFS-Eigenschaften angeben](#).

Amazon VPC

Amazon EMR unterstützt das Starten von Clustern in einer Virtual Private Cloud (VPC) in Amazon VPC. Eine VPC ist ein isoliertes, virtuelles Netzwerk, AWS das die Möglichkeit bietet, erweiterte Aspekte der Netzwerkkonfiguration und des Netzwerkzugriffs zu steuern. Weitere Informationen finden Sie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#).

AWS CloudTrail

Amazon EMR lässt sich integrieren CloudTrail , um Informationen über Anfragen zu protokollieren, die von oder im Namen Ihres AWS Kontos gestellt wurden. Anhand dieser Informationen können Sie verfolgen, wer wann auf Ihr Cluster zugreift sowie die IP-Adresse, von der die Anforderung gestellt wird. Weitere Informationen finden Sie unter [Protokollieren AWS EMR EMR-API-Aufrufen mit AWS CloudTrail](#).

EC2 Amazon-Schlüsselpaare

Indem Sie eine sichere Verbindung zwischen Ihrem Remotecomputer und dem Primärknoten herstellen, können Sie Ihren Cluster überwachen und damit interagieren. Sie verwenden das Netzwerkprotokoll Secure Shell (SSH) für diese Verbindung oder Kerberos für die Authentifizierung. Wenn Sie SSH verwenden, ist ein EC2 Amazon-Schlüsselpaar erforderlich. Weitere Informationen finden Sie unter [Verwenden Sie ein EC2 key pair für SSH-Anmeldeinformationen für Amazon EMR](#).

Überwachen

Sie können die Amazon-EMR-Management-Schnittstellen und Protokolldateien verwenden, um Probleme mit dem Cluster zu beheben, z. B. bei Ausfällen oder Fehlern. Amazon EMR bietet die Möglichkeit, Protokolldateien in Amazon S3 zu archivieren, sodass Sie Protokolle speichern und Probleme beheben können, auch nachdem der Cluster beendet wurde. Amazon EMR bietet in der Amazon-EMR-Konsole auch ein optionales Debugging-Tool, mit dem Sie die Protokolldateien im Hinblick auf Schritte, Aufträge und Aufgaben durchsuchen können. Weitere Informationen finden Sie unter [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#).

Amazon EMR lässt sich integrieren CloudWatch , um Leistungskennzahlen für den Cluster und Jobs innerhalb des Clusters nachzuverfolgen. Sie können Alarme im Hinblick auf eine Vielzahl von Metriken konfigurieren, z. B. ob der Cluster inaktiv ist oder wie viel Prozent des Speicherplatzes verbraucht wurden. Weitere Informationen finden Sie unter [Überwachung von Amazon EMR-Metriken mit CloudWatch](#).

Verwaltungsschnittstellen

Es gibt mehrere Möglichkeiten, mit Amazon EMR zu interagieren:

- Konsole – eine grafische Benutzerschnittstelle, die Sie verwenden können, um Clusters zu starten oder zu verwalten. Hier füllen Sie Webformulare aus, um Detaildaten zum Starten von Clusters anzugeben, Detaildaten von vorhandenen Clusters anzuzeigen und Clusters zu debuggen bzw. zu beenden. Die Konsole bietet die einfachste Möglichkeit für die ersten Schritte mit Amazon EMR keine Programmierkenntnisse erforderlich. Die Konsole ist zu <https://console.aws.amazon.com/elasticmapreduce/Hause> online verfügbar.
- AWS Command Line Interface (AWS CLI) — Eine Client-Anwendung, die Sie auf Ihrem lokalen Computer ausführen, um eine Verbindung zu Amazon EMR herzustellen und Cluster zu erstellen und zu verwalten. Das AWS CLI enthält eine Reihe von Befehlen mit vielen Funktionen, die speziell für Amazon EMR gelten. Damit schreiben Sie Skripts, die das Starten und Verwalten der Clusters

automatisieren. Wenn Sie lieber von einer Befehlszeile aus arbeiten möchten, AWS CLI ist die Verwendung von die beste Option. Weitere Informationen und Beispiele finden Sie unter [Amazon EMR](#) in der AWS CLI -Befehlsreferenz.

- **Software Development Kit (SDK)** — SDKs stellt Funktionen bereit, die Amazon EMR aufrufen, um Cluster zu erstellen und zu verwalten. Mit ihnen können Sie Anwendungen schreiben, die das Erstellen und Verwalten von Clusters automatisieren. Die Verwendung des SDK ist die beste Option, wenn Sie die Funktionen von Amazon EMR erweitern oder anpassen möchten. Amazon EMR ist derzeit in den folgenden Versionen verfügbar SDKs: Go, Java, .NET (C# und VB.NET), Node.js, PHP, Python und Ruby. Weitere Informationen dazu finden Sie SDKs unter [Tools for AWS](#) und [Amazon EMR-Beispielcode und Bibliotheken](#).
- **Web Service API** – eine Low-Level-Schnittstelle, die Sie benutzen können, um den Webservice direkt mithilfe von JSON aufzurufen. Die Verwendung der API ist die beste Option, wenn Sie ein eigenes SDK erstellen wollen, das Amazon EMR aufruft. Weitere Informationen finden Sie in der [Amazon-EMR-API-Referenz](#).

Amazon EMR-Architektur und Service-Ebenen

Die Service-Architektur von Amazon EMR besteht aus mehreren Ebenen, die dem Cluster jeweils bestimmte Möglichkeiten und Funktionen bereitstellen. Dieser Abschnitt bietet eine Übersicht über die jeweiligen Ebenen und Komponenten.

In diesem Thema

- [Speicher](#)
- [Cluster-Ressourcenverwaltung](#)
- [Datenverarbeitungs-Frameworks](#)
- [Anwendungen und Programme](#)

Speicher

Die Speicherschicht umfasst die verschiedenen Dateisysteme, die Sie in Ihrem Cluster verwendet werden. Es gibt mehrere verschiedene Speicheroptionen wie nachfolgend beschrieben.

Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS) ist ein verteiltes, skalierbares Dateisystem für Hadoop. HDFS verteilt die auf verschiedenen Instances im Cluster gespeicherten Daten, wobei mehrere

Kopien von Daten auf unterschiedlichen Instances gespeichert werden, um sicherzustellen, dass bei Ausfall einer einzelnen Instance keine Daten verloren gehen. HDFS ist flüchtiger Speicher, der zurückgefordert wird, wenn Sie einen Cluster beenden. HDFS ist nützlich für das Zwischenspeichern von Zwischenergebnissen während der MapReduce Verarbeitung oder für Workloads mit erheblichen zufälligen I/O-Vorgängen.

Weitere Informationen finden Sie unter [Speicheroptionen und Verhalten von Instances in Amazon EMR](#) im [HDFS-Benutzerhandbuch](#) auf der Website von Apache Hadoop.

EMR File System (EMRFS)

Amazon EMR erweitert mittels des EMR File System (EMRFS) Hadoop durch die Hinzufügung des direkten Zugriffs auf in Amazon S3 gespeicherte Daten, als ob es sich um ein Dateisystem wie HDFS handeln würde. Sie können entweder HDFS oder Amazon S3 als das Dateisystem Ihres Clusters verwenden. In der Regel wird Amazon S3 zum Speichern der Ein- und Ausgabedaten verwendet, Zwischenergebnisse werden in HDFS gespeichert.

Lokales Dateisystem

Das lokale Dateisystem bezieht sich auf einen lokal verbundenen Datenträger. Wenn Sie einen Hadoop-Cluster erstellen, wird jeder Knoten aus einer EC2 Amazon-Instance erstellt, die über einen vorkonfigurierten Block von vorinstalliertem Festplattenspeicher verfügt, der als Instance-Speicher bezeichnet wird. Daten zu Instance-Speicher-Volumes bleiben nur während des Lebenszyklus der EC2 Amazon-Instance erhalten.

Cluster-Ressourcenverwaltung

Der Ressourcenverwaltungs-Layer ist verantwortlich für die Verwaltung der Cluster-Ressourcen und die Planung der Aufträge für die Datenverarbeitung.

Amazon EMR verwendet standardmäßig YARN (Yet Another Resource Negotiator). Dabei handelt es sich um eine Komponente, die in Apache Hadoop 2.0 eingeführt wurde und mit der die Cluster-Ressourcen für mehrere Datenverarbeitungs-Frameworks zentral verwaltet werden können. Es gibt jedoch auch andere Frameworks und Anwendungen, die in Amazon EMR bereitgestellt werden und nicht YARN als Ressourcenmanager verwenden. Amazon EMR verfügt außerdem auf jedem Knoten, der YARN-Komponenten verwaltet, über einen Agenten, der den Cluster stabil erhält und mit dem Amazon-EMR-Service kommuniziert.

Da Spot Instances häufig zum Ausführen von Aufgabenknoten verwendet werden, verfügt Amazon EMR über Standardfunktionen für die Planung von YARN-Aufträgen, sodass laufende Aufträge nicht

fehlschlagen, wenn Aufgabenknoten, die auf Spot Instances ausgeführt werden, beendet werden. Amazon EMR ermöglicht dies, indem Anwendungsmasterprozesse nur auf Core-Knoten ausgeführt werden können. Der Anwendungsmasterprozess steuert die Ausführung von Aufträgen und muss während der gesamten Laufzeit des Auftrags aktiv bleiben.

Amazon-EMR-Version 5.19.0 und höher verwendet zu diesem Zweck das integrierte [YARN-Knotenbeschriftungsfeature](#). (Frühere Versionen verwendeten einen Code-Patch). Die Eigenschaften in den Klassifizierungen `yarn-site` und in der `capacity-scheduler`-Konfiguration sind standardmäßig so konfiguriert, dass der YARN-Kapazitätsplaner und der Fair-Scheduler die Vorteile von Knotenbezeichnungen nutzen. Amazon EMR kennzeichnet Core-Knoten automatisch mit dem `CORE`-Label und legt Eigenschaften fest, sodass Anwendungsmaster nur für Knoten mit dem `CORE`-Label geplant werden. Durch manuelles Ändern verwandter Eigenschaften in den Konfigurationsklassifizierungen von Yarn-Site und Kapazitätsplaner oder direkt in den zugehörigen XML-Dateien könnte diese Feature beeinträchtigt oder verändert werden.

Datenverarbeitungs-Frameworks

Der Datenverarbeitungs-Framework-Layer ist die Engine, die zur Verarbeitung und Analyse der Daten verwendet wird. Es stehen viele Frameworks zur Verfügung, die auf YARN ausgeführt werden oder über ihre eigene Ressourcenverwaltung verfügen. Es gibt unterschiedliche Frameworks für die verschiedenen Verarbeitungsanforderungen, beispielsweise Stapel, Interaktiv, In-Memory, Streaming und so weiter. Das Framework, das Sie auswählen sollten, hängt von Ihrem Anwendungsfall ab. Dies wirkt sich auf die Sprachen und Schnittstellen der Anwendungsebene aus, d. h. der Ebene, über die mit den zu verarbeitenden Daten interagiert wird. Die wichtigsten für Amazon EMR verfügbaren Verarbeitungs-Frameworks sind Hadoop MapReduce und Spark.

Hadoop MapReduce

Hadoop MapReduce ist ein Open-Source-Programmiermodell für verteiltes Rechnen. Es vereinfacht den Prozess der Entwicklung paralleler verteilter Anwendungen, indem die gesamte Logik gehandhabt wird, während Sie die Funktionen "Map" und "Reduce" bereitstellen. Die Funktion "Map" führt eine Zuordnung von Daten und Sätzen von Schlüssel/Wert-Paaren durch, die als Zwischenergebnisse bezeichnet werden. Die Funktion "Reduce" kombiniert die Zwischenergebnisse, wendet weitere Algorithmen an und generiert das Endergebnis. Es stehen mehrere Frameworks zur Verfügung MapReduce, z. B. Hive, das automatisch Map- und Reduce-Programme generiert.

Weitere Informationen finden Sie unter [Wie Karten- und Reduziervorgänge tatsächlich ausgeführt werden](#) auf der Wiki-Website von Apache Hadoop.

Apache Spark

Spark ist ein Cluster-Framework und Programmiermodell für die Verarbeitung von Big-Data-Workloads. Wie Hadoop MapReduce ist Spark ein verteiltes Open-Source-Verarbeitungssystem, verwendet jedoch gerichtete azyklische Graphen für Ausführungspläne und In-Memory-Caching für Datensätze. Wenn Sie Spark auf Amazon EMR ausführen, können Sie über EMRFS direkt auf Ihre Daten in Amazon S3 zugreifen. Spark unterstützt mehrere interaktive Abfragen Module wie beispielsweise SparkSQL.

Weitere Informationen finden Sie unter [Apache Spark in Amazon-EMR-Clusters](#) in den Amazon-EMR-Versionshinweise.

Anwendungen und Programme

Amazon EMR unterstützt zahlreiche Anwendungen, wie Hive, Pig, und die Spark Streaming-Bibliothek, um beispielsweise mithilfe komplexerer Programmiersprachen Verarbeitungs-Workloads zu erstellen, Machine-Learning-Algorithmen zu nutzen, Anwendungen für die Stream-Verarbeitung zu erstellen und Data Warehouses zu entwickeln. Darüber hinaus unterstützt Amazon EMR auch Open-Source-Projekte, die ihre eigene Cluster-Management-Funktionalität mitbringen und nicht YARN verwenden.

Sie können verschiedene Bibliotheken und Sprachen verwenden, um mit den Anwendungen, die Sie in Amazon EMR ausführen, zu interagieren. Sie können beispielsweise Java, Hive oder Pig mit MapReduce oder Spark Streaming, Spark SQL und GraphX mit Spark verwenden. MLlib

Weitere Informationen finden Sie im [Handbuch zu Amazon-EMR-Versionen](#).

Bevor Sie Amazon EMR einrichten

Führen Sie die in diesem Abschnitt beschriebenen vorbereitenden Aufgaben durch, bevor Sie einen Amazon EMR-Cluster zum ersten Mal starten. Dazu gehören die Einrichtung Ihres AWS Kontos, falls Sie eines benötigen, und die Durchführung von Maßnahmen zur Einrichtung einer sicheren Kommunikation.

Melde dich an für ein AWS-Konto

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

Um sich für eine anzumelden AWS-Konto

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

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

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

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

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

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

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

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

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

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

Anmelden als Administratorbenutzer

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

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

Weiteren Benutzern Zugriff zuweisen

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

Anweisungen hierzu finden Sie unter [Berechtigungssatz erstellen](#) im AWS IAM Identity Center Benutzerhandbuch.

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

Eine genaue Anleitung finden Sie unter [Gruppen hinzufügen](#) im AWS IAM Identity Center Benutzerhandbuch.

Erstellen Sie ein EC2 Amazon-Schlüsselpaar für SSH

Note

Mit Amazon-EMR-Version 5.10.0 oder höher können Sie Kerberos zur Authentifizierung von Benutzern und SSH-Verbindungen zu einem Cluster konfigurieren. Weitere Informationen finden Sie unter [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#).

Um die Knoten in einem Cluster über einen sicheren Kanal mithilfe des Secure Shell (SSH) - Protokolls zu authentifizieren und eine Verbindung zu ihnen herzustellen, erstellen Sie ein Amazon Elastic Compute Cloud (Amazon EC2) -Schlüsselpaar, bevor Sie den Cluster starten. Außerdem können Sie auch einen Cluster ohne ein Schlüsselpaar erstellen. Dies geschieht normalerweise mit vorübergehenden Clustern, die starten, gewisse Schritte ausführen und dann automatisch beendet werden.

Wenn ...	Dann ...
Sie haben bereits ein EC2 Amazon-Schlüsselpaar, das Sie verwenden möchten, oder Sie müssen sich nicht bei Ihrem Cluster authentifizieren.	Überspringen Sie diesen Schritt.
Sie müssen ein Schlüsselpaar erstellen.	Weitere Informationen finden Sie unter Erstellen Ihres key pair mit Amazon EC2 .

Nächste Schritte

- Hinweise zur Erstellung eines Beispielclusters finden Sie unter [Tutorial: Erste Schritte mit Amazon EMR](#).
- Weitere Informationen zur Konfiguration eines benutzerdefinierten Clusters und zur Steuerung des Zugriffs darauf finden Sie unter [Planung, Konfiguration und Start von Amazon EMR-Clustern](#) und [Sicherheit in Amazon EMR](#).

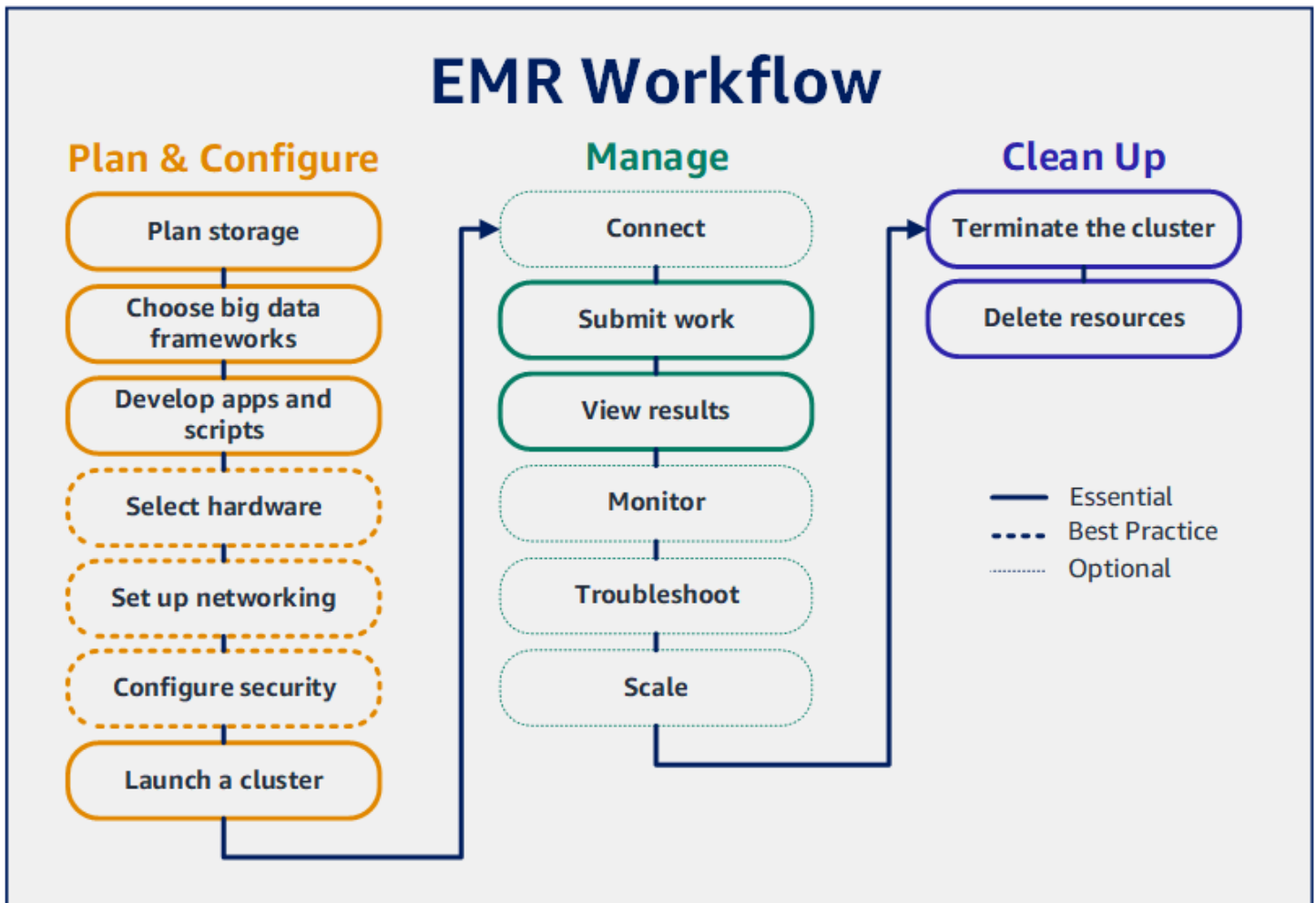
Tutorial: Erste Schritte mit Amazon EMR

Gehen Sie durch einen Workflow, um schnell einen Amazon EMR-Cluster einzurichten und eine Spark-Anwendung auszuführen.

Einrichtung Ihres Amazon EMR-Clusters

Mit Amazon EMR können Sie in nur wenigen Minuten einen Cluster einrichten, um Daten mit Big-Data-Frameworks zu verarbeiten und zu analysieren. Dieses Tutorial zeigt Ihnen, wie Sie einen Beispielcluster mit Spark starten und wie Sie ein einfaches PySpark Skript ausführen, das in einem Amazon S3 S3-Bucket gespeichert ist. Es behandelt wichtige Amazon-EMR-Aufgaben in drei Workflow-Hauptkategorien: Planen und Konfigurieren, Verwalten und Aufräumen.

Während Sie das Tutorial durcharbeiten, finden Sie Links zu detaillierteren Themen und im [Nächste Schritte](#) Abschnitt Ideen für weitere Schritte. Bei weiteren Fragen oder Problemen können Sie sich an das Amazon-EMR-Team wenden, indem Sie einen Beitrag im [Diskussionsforum](#) veröffentlichen.



Voraussetzungen

- Stellen Sie vor dem Starten eines Amazon-EMR-Clusters sicher, dass Sie die Aufgaben unter [Bevor Sie Amazon EMR einrichten](#) ausführen.

Kosten

- Der erstellte Beispiel-Cluster wird in einer Live-Umgebung ausgeführt. Für den Cluster fallen nur minimale Gebühren an. Stellen Sie sicher, dass Sie die Bereinigungsaufgaben im letzten Schritt dieses Tutorials ausführen, um zusätzliche Kosten zu vermeiden. Gebühren fallen pro Sekunde gemäß den Amazon-EMR-Preisen an. Die Gebühren variieren auch je nach Region. Weitere Informationen finden Sie unter [Amazon-EMR-Preise](#).
- Für kleine Dateien, die Sie in Amazon S3 speichern, können geringe Gebühren anfallen. Einige oder alle Gebühren für Amazon S3 können erlassen werden, wenn Sie die Nutzungsgrenzen des

AWS kostenlosen Kontingents einhalten. Weitere Informationen finden Sie unter [Amazon-S3-Preise](#) und [AWS kostenloses Kontingent](#).

Schritt 1: Datenressourcen konfigurieren und einen Amazon EMR-Cluster starten

Speicher für Amazon EMR vorbereiten

Wenn Sie Amazon EMR verwenden, können Sie aus einer Vielzahl von Dateisystemen wählen, um Eingabedaten, Ausgabedaten und Protokolldateien zu speichern. In diesem Tutorial verwenden Sie EMRFS zum Speichern von Daten in einem S3-Bucket. EMRFS ist eine Implementierung des Hadoop-Dateisystems, mit der Sie reguläre Dateien in Amazon S3 lesen und schreiben können. Weitere Informationen finden Sie unter [Arbeiten mit Speicher- und Dateisystemen mit Amazon EMR](#).

Um einen Bucket zu erstellen, befolgen Sie die Anweisungen unter [Wie wird ein S3 Bucket erstellt?](#) im Konsolen-Benutzerhandbuch zu Amazon Simple Storage Service. Erstellen Sie den Bucket in derselben AWS Region, in der Sie Ihren Amazon EMR-Cluster starten möchten. Zum Beispiel USA West (Oregon) us-west-2.

Für Buckets und Ordner, die Sie mit Amazon EMR verwenden, gelten die folgenden Einschränkungen:

- Namen können Kleinbuchstaben, Zahlen, Bindestriche (-) und Punkte (.) enthalten.
- Namen dürfen nicht mit Zahlen enden.
- Bucket-Namen müssen in allen AWS -Konten eindeutig sein.
- Ein Ausgabeordner muss leer sein.

Bereiten Sie eine Anwendung mit Eingabedaten für Amazon EMR vor

Die gängigste Methode zur Vorbereitung eines Antrags für Amazon EMR besteht darin, den Antrag und seine Eingabedaten in Amazon S3 hochzuladen. Wenn Sie dann Arbeit an Ihren Cluster senden, geben Sie die Amazon-S3-Speicherorte für Ihr Skript und Ihre Daten an.

In diesem Schritt laden Sie ein PySpark Beispielskript in Ihren Amazon S3 S3-Bucket hoch. Wir haben ein PySpark Skript bereitgestellt, das Sie verwenden können. Das Skript verarbeitet Inspektionsdaten von Lebensmittelbetrieben und gibt eine Ergebnisdatei in Ihrem S3-Bucket

zurück. In der Ergebnisdatei sind die zehn Einrichtungen mit den meisten Verstößen vom Typ „Rot“ aufgeführt.

Sie laden auch Beispieleingabedaten in Amazon S3 hoch, damit das PySpark Skript sie verarbeiten kann. Bei den Eingabedaten handelt es sich um eine modifizierte Version der Inspektionsergebnisse des Gesundheitsministeriums in King County, Washington, von 2006 bis 2020. Weitere Informationen finden Sie unter [King County Open Data: Daten zur Inspektion von Lebensmittelbetrieben](#).
Nachfolgend sehen Sie einige Beispielzeilen aus dem Datensatz.

```
name, inspection_result, inspection_closed_business, violation_type, violation_points
100 LB CLAM, Unsatisfactory, FALSE, BLUE, 5
100 PERCENT NUTRICION, Unsatisfactory, FALSE, BLUE, 5
7-ELEVEN #2361-39423A, Complete, FALSE, , 0
```

Um das PySpark Beispielskript für EMR vorzubereiten

1. Kopieren Sie den Beispielcode unten mit einem Editor Ihrer Wahl in eine neue Datei.

```
import argparse

from pyspark.sql import SparkSession

def calculate_red_violations(data_source, output_uri):
    """
    Processes sample food establishment inspection data and queries the data to
    find the top 10 establishments
    with the most Red violations from 2006 to 2020.

    :param data_source: The URI of your food establishment data CSV, such as 's3://
    amzn-s3-demo-bucket/food-establishment-data.csv'.
    :param output_uri: The URI where output is written, such as 's3://amzn-s3-demo-
    bucket/restaurant_violation_results'.
    """
    with SparkSession.builder.appName("Calculate Red Health
    Violations").getOrCreate() as spark:
        # Load the restaurant violation CSV data
        if data_source is not None:
            restaurants_df = spark.read.option("header", "true").csv(data_source)

        # Create an in-memory DataFrame to query
        restaurants_df.createOrReplaceTempView("restaurant_violations")
```

```
# Create a DataFrame of the top 10 restaurants with the most Red violations
top_red_violation_restaurants = spark.sql("""SELECT name, count(*) AS
total_red_violations
FROM restaurant_violations
WHERE violation_type = 'RED'
GROUP BY name
ORDER BY total_red_violations DESC LIMIT 10""")

# Write the results to the specified output URI
top_red_violation_restaurants.write.option("header",
"true").mode("overwrite").csv(output_uri)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '--data_source', help="The URI for you CSV restaurant data, like an S3
bucket location.")
    parser.add_argument(
        '--output_uri', help="The URI where output is saved, like an S3 bucket
location.")
    args = parser.parse_args()

    calculate_red_violations(args.data_source, args.output_uri)
```

2. Speichern Sie die Datei als `health_violations.py`.
3. Laden Sie Ihre `health_violations.py` in Amazon S3 in den Bucket hoch, den Sie als Voraussetzung für dieses Tutorial erstellt haben. Anweisungen finden Sie unter [Hochladen eines Objekts in Ihren Bucket](#) im Handbuch „Erste Schritte“ für Amazon Simple Storage Service.

Um die Probeneingabedaten für EMR vorzubereiten

1. Laden Sie die ZIP-Datei [food_establishment_data.zip](#) herunter.
2. Entpacken und speichern Sie `food_establishment_data.zip` als `food_establishment_data.csv` auf Ihrem Computer.
3. Laden Sie die CSV-Datei in den S3-Bucket hoch, den Sie für dieses Tutorial erstellt haben. Anweisungen finden Sie unter [Hochladen eines Objekts in Ihren Bucket](#) im Handbuch „Erste Schritte“ für Amazon Simple Storage Service.

Weitere Informationen zum Einrichten von Daten für EMR finden Sie unter [Eingabedaten für die Verarbeitung mit Amazon EMR vorbereiten](#).

Starten eines Amazon-EMR-Clusters

Nachdem Sie einen Speicherort und Ihre Anwendung vorbereitet haben, können Sie einen Amazon-EMR-Beispielcluster starten. In diesem Schritt starten Sie einen Apache-Spark-Cluster mit der neuesten [Amazon-EMR-Version](#).

Console

Um einen Cluster zu starten, auf dem Spark zusammen mit der Konsole installiert ist

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Notieren Sie sich auf der Seite Cluster erstellen die Standardwerte für Version, Instance-Typ, Anzahl der Instances und Berechtigungen. Diese Felder werden automatisch mit Werten aufgefüllt, die für Allzweck-Cluster geeignet sind.
4. Geben Sie im Feld Clusternamen einen eindeutigen Clusternamen ein, um Ihren Cluster leichter identifizieren zu können, z. B. *My first cluster* Ihr Clusternamen darf die Zeichen <, >, \$, | oder ` (Backtick) nicht enthalten.
5. Wählen Sie unter Anwendungen die Spark-Option, um Spark auf Ihrem Cluster zu installieren.

Note

Wählen Sie die Anwendungen aus, die Sie in Ihrem Amazon-EMR-Cluster haben möchten, bevor Sie den Cluster starten. Sie können nach dem Start keine Anwendungen zu einem Cluster hinzufügen oder daraus entfernen.

6. Aktivieren Sie unter Cluster-Protokolle das Kontrollkästchen Cluster-spezifische Protokolle in Amazon S3 veröffentlichen. Ersetzen Sie den Amazon-S3-Standortwert durch den Amazon-S3-Bucket, den Sie erstellt haben, gefolgt von **/logs**. Beispiel, **s3://amzn-s3-demo-bucket/logs**. Durch das Hinzufügen von **/logs** wird ein neuer Ordner namens „logs“ in Ihrem Bucket erstellt, in den Amazon EMR die Protokolldateien Ihres Clusters kopieren kann.

7. Wählen Sie unter Sicherheitskonfiguration und Berechtigungen Ihr EC2 key pair aus. Wählen Sie im selben Abschnitt das Dropdownmenü Servicerolle für Amazon EMR aus und wählen Sie EMR_ aus. DefaultRole Wählen Sie dann das Dropdownmenü „IAM-Rolle für das Instanzprofil“ und dann „EMR_“ aus. EC2 DefaultRole
8. Wählen Sie Cluster erstellen aus, um den Cluster zu starten und die Cluster-Detailseite zu öffnen.
9. Suchen Sie den Cluster-Status neben dem Clusternamen. Der Status ändert sich von Starten zu Läuft zu Wartend, wenn Amazon EMR den Cluster bereitstellt. Möglicherweise müssen Sie das Aktualisierungs-Symbol auf der rechten Seite betätigen oder Ihren Browser aktualisieren, um Updates zu sehen.

Ihr Clusterstatus ändert sich in Wartend, wenn der Cluster betriebsbereit ist, läuft und bereit ist, Arbeit anzunehmen. Weitere Informationen zum Lesen der Cluster-Zusammenfassung finden Sie unter [Status und Details des Amazon EMR-Clusters anzeigen](#). Weitere Informationen zu Cluster-Status finden Sie unter [Verstehen des Cluster-Lebenszyklus](#).

CLI

Um einen Cluster zu starten, auf dem Spark mit dem installiert ist AWS CLI

1. Erstellen Sie IAM-Standardrollen, die Sie dann verwenden können, um Ihren Cluster zu erstellen, indem Sie den folgenden Befehl verwenden.

```
aws emr create-default-roles
```

Weitere Informationen zu create-default-roles finden Sie in der [AWS CLI - Befehlsreferenz](#).

2. Erstellen Sie einen Spark-Cluster mit dem folgenden Befehl. Geben Sie mit der --name Option einen Namen für Ihren Cluster ein und geben Sie mit der --ec2-attributes Option den Namen Ihres EC2 key pair an.

```
aws emr create-cluster \  
--name "<My First EMR Cluster>" \  
--release-label <emr-5.36.2> \  
--applications Name=Spark \  
--ec2-attributes KeyName=<myEMRKeyName> \  
--instance-type m5.xlarge \  
--instance-count 3 \  

```

```
--use-default-roles
```

Notieren Sie sich die anderen erforderlichen Werte für `--instance-type`, `--instance-count` und `--use-default-roles`. Diese Werte wurden für Allzweck-Cluster ausgewählt. Weitere Informationen zu `create-cluster` finden Sie in der [AWS CLI -Befehlsreferenz](#).

 Note

Linux-Zeilenfortsetzungszeichen (`\`) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (`^`).

Die Ausgabe sollte ungefähr wie die folgende aussehen. Die Ausgabe zeigt `ClusterId` und `ClusterArn` Ihres neuen Clusters. Notieren Sie sich Ihre `ClusterId`. Sie verwenden `ClusterId`, um den Clusterstatus zu überprüfen und Arbeiten einzureichen.

```
{
  "ClusterId": "myClusterId",
  "ClusterArn": "myClusterArn"
}
```

- Überprüfen Sie Ihren Clusterstatus mit dem folgenden Befehl.

```
aws emr describe-cluster --cluster-id <myClusterId>
```

Mit dem Status-Objekt für Ihren neuen Cluster sollten Sie eine Ausgabe wie die folgende sehen.

```
{
  "Cluster": {
    "Id": "myClusterId",
    "Name": "My First EMR Cluster",
    "Status": {
      "State": "STARTING",
      "StateChangeReason": {
        "Message": "Configuring cluster software"
      }
    }
  }
}
```

```
}
```

Der State-Wert ändert sich von `STARTING` zu `RUNNING` zu `WAITING`, wenn Amazon EMR den Cluster bereitstellt.

Der Cluster-Status ändert sich zu **WAITING**, in dem ein Cluster betriebsbereit und bereit ist, Arbeit anzunehmen. Weitere Informationen zu Cluster-Status finden Sie unter [Verstehen des Cluster-Lebenszyklus](#).

Schritt 2: Arbeiten an Ihren Amazon EMR-Cluster einreichen

Arbeit einreichen und Ergebnisse ansehen

Nachdem Sie einen Cluster gestartet haben, können Sie Arbeiten an den laufenden Cluster senden, um Daten zu verarbeiten und zu analysieren. In einem Schritt reichen Sie Arbeiten an einen Amazon-EMR-Cluster ein. Ein Schritt ist eine Arbeitseinheit, die aus einer oder mehreren Aktionen besteht. Sie könnten beispielsweise einen Schritt zur Berechnung von Werten oder zur Übertragung und Verarbeitung von Daten einreichen. Sie können Schritte beim Erstellen eines Clusters oder an einen laufenden Cluster senden. In diesem Teil des Tutorials übermitteln Sie `health_violations.py` als Schritt an Ihren laufenden Cluster. Weitere Informationen zu Schritten finden Sie unter [Arbeit an einen Amazon EMR-Cluster einreichen](#).

Console

Um eine Spark-Anwendung als Schritt mit der Konsole einzureichen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters aus, und wählen Sie dann den Cluster aus, für den Sie Arbeit einreichen möchten. Der Clusterstatus muss Wartend lauten.
3. Wählen Sie Schritte und dann Schritt hinzufügen.
4. Konfigurieren Sie den Schritt anhand der folgenden Richtlinien:
 - Wählen Sie für Typ die Option Spark-Anwendung aus. Sie sollten zusätzliche Felder für den Bereitstellungsmodus, den Speicherort der Anwendung und die Optionen Spark-Submit sehen.

- Geben Sie unter Name einen neuen Namen ein. Wenn Sie viele Schritte in einem Cluster haben, hilft Ihnen die Benennung der einzelnen Schritte dabei, den Überblick zu behalten.
- Behalten Sie für den Bereitstellungsmodus den Standardwert Clustermodus bei. Weitere Informationen zu Spark-Bereitstellungsmodi finden Sie unter [Übersicht über den Clustermodus](#) in der Apache-Spark-Dokumentation.
- Geben Sie unter Anwendungsort den Speicherort Ihres `health_violations.py` Skripts in Amazon S3 ein, z. `s3://amzn-s3-demo-bucket/health_violations.py` B.
- Lassen Sie das Feld mit den Spark-Submit-Optionen leer. Weitere Informationen zu den `spark-submit`-Optionen finden Sie unter [Starten von Anwendungen mit spark-submit](#).
- Geben Sie im Feld Argumente die folgenden Argumente und Werte ein:

```
--data_source s3://amzn-s3-demo-bucket/food_establishment_data.csv  
--output_uri s3://amzn-s3-demo-bucket/myOutputFolder
```

`s3://amzn-s3-demo-bucket/food_establishment_data.csv` Ersetzen Sie es durch die S3-Bucket-URI der Eingabedaten, in denen Sie sie vorbereitet haben [Bereiten Sie eine Anwendung mit Eingabedaten für Amazon EMR vor](#).

`amzn-s3-demo-bucket` Ersetzen Sie ihn durch den Namen des Buckets, den Sie für dieses Tutorial erstellt haben, und `myOutputFolder` ersetzen Sie ihn durch einen Namen für Ihren Cluster-Ausgabeordner.

- Übernehmen Sie unter Aktion bei Fehler des Schrittes die Standardeinstellung Fortfahren. Auf diese Weise wird der Cluster weiter ausgeführt, wenn der Schritt fehlschlägt.
5. Wählen Sie Hinzufügen, um den Schritt zu senden. Der Schritt wird in der Konsole mit dem Status Ausstehend angezeigt.
 6. Überwachen Sie den Status des Schritts. Der Wert sollte sich von Ausstehend zu Wird ausgeführt zu Abgeschlossen ändern. Um den Status in der Konsole zu aktualisieren, wählen Sie das Aktualisierungssymbol rechts neben dem Filter aus. Die Ausführung des Skripts dauert etwa eine Minute. Wenn sich der Status in Abgeschlossen, ändert, wurde der Schritt erfolgreich abgeschlossen.

CLI

Um eine Spark-Anwendung als Schritt einzureichen, verwenden Sie AWS CLI

1. Stellen Sie sicher, dass Sie ClusterId des Clusters haben, den Sie in [Starten eines Amazon-EMR-Clusters](#) gestartet haben. Sie können Ihre Cluster-ID auch mit dem folgenden Befehl abrufen.

```
aws emr list-clusters --cluster-states WAITING
```

2. Senden Sie health_violations.py als Schritt mit dem add-steps-Befehl und Ihrem ClusterId.
 - Sie können einen Namen für Ihren Schritt angeben, indem Sie ihn ersetzen *"My Spark Application"*. Ersetzen Sie im Args Array *s3://amzn-s3-demo-bucket/health_violations.py* durch den Speicherort Ihrer health_violations.py Anwendung.
 - *s3://amzn-s3-demo-bucket/food_establishment_data.csv* Ersetzen Sie durch den S3-Speicherort Ihres food_establishment_data.csv Datensatzes.
 - *s3://amzn-s3-demo-bucket/MyOutputFolder* Ersetzen Sie durch den S3-Pfad Ihres angegebenen Buckets und einen Namen für Ihren Cluster-Ausgabeordner.
 - ActionOnFailure=CONTINUE bedeutet, dass der Cluster weiter ausgeführt wird, wenn der Schritt fehlschlägt.

```
aws emr add-steps \  
--cluster-id <myClusterId> \  
--steps Type=Spark,Name="<My Spark  
Application>",ActionOnFailure=CONTINUE,Args=[<s3://amzn-s3-demo-  
bucket/health_violations.py>,<--data_source,<s3://amzn-s3-demo-bucket/  
food_establishment_data.csv>,<--output_uri,<s3://amzn-s3-demo-bucket/  
MyOutputFolder>]
```

Weitere Informationen zum Senden von Schritten mithilfe der CLI finden Sie in der [AWS CLI - Befehlsreferenz](#).

Nachdem Sie den Schritt eingereicht haben, sollten Sie eine Ausgabe wie die folgende mit einer Liste von StepIds sehen. Da Sie einen Schritt eingereicht haben, wird in der Liste nur

eine ID angezeigt. Kopieren Sie Ihre Schritt-ID. Sie verwenden Ihre Schritt-ID, um den Status des Schritts zu überprüfen.

```
{
  "StepIds": [
    "s-1XXXXXXXXXXA"
  ]
}
```

3. Fragen Sie den Status Ihres Schritts mit dem `describe-step`-Befehl ab.

```
aws emr describe-step --cluster-id <myClusterId> --step-id <s-1XXXXXXXXXXA>
```

Die Ausgabe sollte ungefähr wie die folgende aussehen, mit Informationen zu Ihrem Schritt.

```
{
  "Step": {
    "Id": "s-1XXXXXXXXXXA",
    "Name": "My Spark Application",
    "Config": {
      "Jar": "command-runner.jar",
      "Properties": {},
      "Args": [
        "spark-submit",
        "s3://amzn-s3-demo-bucket/health_violations.py",
        "--data_source",
        "s3://amzn-s3-demo-bucket/food_establishment_data.csv",
        "--output_uri",
        "s3://amzn-s3-demo-bucket/myOutputFolder"
      ]
    },
    "ActionOnFailure": "CONTINUE",
    "Status": {
      "State": "COMPLETED"
    }
  }
}
```

Der State-Wert des Schritts ändert sich mit der Ausführung des Schritts von PENDING zu RUNNING zu COMPLETED. Die Ausführung des Schritts dauert etwa eine Minute, sodass Sie den Status möglicherweise einige Male überprüfen müssen.

Sie wissen, dass der Schritt erfolgreich war, wenn sich State in **COMPLETED** ändert.

Weitere Informationen zum Schrittlebenszyklus finden Sie unter [Ausführen von Schritten zur Verarbeitung von Daten](#).

Ergebnisse anzeigen

Nachdem ein Schritt erfolgreich ausgeführt wurde, können Sie seine Ausgabeergebnisse in Ihrem Amazon-S3-Ausgabeordner anzeigen.

So sehen Sie die Ergebnisse von **health_violations.py**

1. Öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Bucket-Namen und dann den Ausgabeordner aus, den Sie beim Absenden des Schritts angegeben haben. Zum Beispiel *amzn-s3-demo-bucket* und dann *myOutputFolder*.
3. Stellen Sie sicher, dass die folgenden Elemente in Ihrem Ausgabeordner angezeigt werden:
 - Ein kleines Objekt namens `_SUCCESS`.
 - Eine CSV-Datei, die mit dem Präfix `part-` beginnt, die Ihre Ergebnisse enthält.
4. Wählen Sie das Objekt mit Ihren Ergebnissen aus und klicken Sie dann auf Herunterladen, um die Ergebnisse in Ihrem lokalen Dateisystem zu speichern.
5. Öffnen Sie die Ergebnisse in Ihrem Editor Ihrer Wahl. In der Ausgabedatei sind die zehn Lebensmittelbetriebe mit den meisten roten Verstößen aufgeführt. Die Ausgabedatei zeigt auch die Gesamtzahl der roten Verstöße für jeden Betrieb.

Es folgt ein Beispiel für ein `health_violations.py`-Ergebnis.

```
name, total_red_violations
SUBWAY, 322
T-MOBILE PARK, 315
WHOLE FOODS MARKET, 299
PCC COMMUNITY MARKETS, 251
TACO TIME, 240
```

```
MCDONALD'S, 177
THAI GINGER, 153
SAFEWAY INC #1508, 143
TAQUERIA EL RINCONSITO, 134
HIMITSU TERIYAKI, 128
```

Weitere Informationen zur Amazon-EMR-Clusterausgabe finden Sie unter [Einen Speicherort für die Amazon EMR-Clusterausgabe konfigurieren](#).

(Optional) Stellen Sie eine Verbindung zu Ihrem laufenden Amazon-EMR-Cluster her

Wenn Sie Amazon EMR verwenden, möchten Sie möglicherweise eine Verbindung zu einem laufenden Cluster herstellen, um Protokolldateien zu lesen, den Cluster zu debuggen oder CLI-Tools wie die Spark-Shell zu verwenden. Mit Amazon EMR können Sie mithilfe des Secure Shell (SSH)-Protokolls eine Verbindung zu einem Cluster herstellen. In diesem Abschnitt erfahren Sie, wie Sie SSH konfigurieren, eine Verbindung zu Ihrem Cluster herstellen und Protokolldateien für Spark anzeigen. Weitere Informationen zum Herstellen einer Verbindung mit einem Cluster finden Sie unter [Authentifizieren von Amazon-EMR-Cluster-Knoten](#).

Autorisieren von SSH-Verbindungen zu Ihrem Cluster

Bevor Sie eine Verbindung zu Ihrem Cluster herstellen, müssen Sie Ihre Cluster-Sicherheitsgruppen ändern, um eingehende SSH-Verbindungen zu autorisieren. EC2 Amazon-Sicherheitsgruppen dienen als virtuelle Firewalls zur Steuerung des ein- und ausgehenden Datenverkehrs zu Ihrem Cluster. Als Sie Ihren Cluster für dieses Tutorial erstellt haben, hat Amazon EMR in Ihrem Namen die folgenden Sicherheitsgruppen erstellt:

ElasticMapReduce-Meister

Die standardmäßige verwaltete Amazon-EMR-Sicherheitsgruppe, die mit dem Primärknoten verknüpft ist. In einem Amazon EMR-Cluster ist der primäre Knoten eine EC2 Amazon-Instance, die den Cluster verwaltet.

ElasticMapReduce-Slave

Die Standardsicherheitsgruppe, die Core- und Aufgabenknoten zugeordnet ist.

Console

Um SSH-Zugriff für vertrauenswürdige Quellen für die primäre Sicherheitsgruppe mit der Konsole zu ermöglichen

Um Ihre Sicherheitsgruppen zu bearbeiten, benötigen Sie die Berechtigung, Sicherheitsgruppen für die VPC zu verwalten, in der sich der Cluster befindet. Weitere Informationen finden Sie unter [Ändern von Benutzerberechtigungen](#) und unter [Beispielrichtlinie](#), die die Verwaltung von EC2 Sicherheitsgruppen ermöglicht, im IAM-Benutzerhandbuch.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten. Dadurch wird die Cluster-Detailseite geöffnet. Die Registerkarte Eigenschaften auf dieser Seite sollte vorausgewählt sein.
3. Wählen Sie auf der Registerkarte Eigenschaften unter Netzwerk den Pfeil neben EC2 Sicherheitsgruppen (Firewall) aus, um diesen Abschnitt zu erweitern. Wählen Sie unter Primärknoten den Link zur Sicherheitsgruppe aus. Wenn Sie die folgenden Schritte abgeschlossen haben, können Sie optional zu diesem Schritt zurückkehren, Core- und Aufgabenknoten auswählen und die folgenden Schritte wiederholen, um dem SSH-Client Zugriff auf Core- und Aufgabenknoten zu gewähren.
4. Dadurch wird die EC2 Konsole geöffnet. Wählen Sie die Registerkarte Eingehende Regeln und anschließend Eingehende Regeln bearbeiten aus.
5. Suchen Sie mit den folgenden Einstellungen nach einer Regel für eingehenden Datenverkehr, die öffentlichen Zugriff ermöglicht. Falls sie existiert, wählen Sie Löschen, um sie zu entfernen.

- Typ

SSH

- Port

22

- Quelle

Benutzerdefiniert 0.0.0.0/0

 Warning

Vor Dezember 2020 verfügte die Sicherheitsgruppe ElasticMapReduce -master über eine vorkonfigurierte Regel, die eingehenden Datenverkehr auf Port 22 aus allen Quellen zuließ. Diese Regel wurde erstellt, um die ersten SSH-Verbindungen zum Hauptknoten zu vereinfachen. Wir empfehlen Ihnen dringend, diese Eingangsregel zu entfernen und den Datenverkehr auf vertrauenswürdige Quellen zu beschränken.

6. Scrollen Sie zum Ende der Regelliste und wählen Sie Regel hinzufügen.
7. Wählen Sie für Type (Typ) SSH aus. Wenn Sie SSH auswählen, wird automatisch TCP für Protokoll und 22 für Portbereich eingegeben.
8. Wählen Sie als Quelle Meine IP aus, um Ihre IP-Adresse automatisch als Quelladresse hinzuzufügen. Sie können auch einen Bereich benutzerdefinierter vertrauenswürdiger Client-IP-Adressen hinzufügen oder zusätzliche Regeln für andere Clients erstellen. In vielen Netzwerkumgebungen werden IP-Adressen dynamisch zugewiesen, sodass Sie in Zukunft möglicherweise Ihre IP-Adressen für vertrauenswürdige Clients aktualisieren müssen.
9. Wählen Sie Speichern.
10. Wählen Sie optional Core- und Aufgabenknoten aus der Liste aus und wiederholen Sie die obigen Schritte, um dem SSH-Client Zugriff auf Core- und Aufgabenknoten zu gewähren.

Connect zu Ihrem Cluster her, indem Sie AWS CLI

Unabhängig von Ihrem Betriebssystem können Sie mit dem AWS CLI eine SSH-Verbindung zu Ihrem Cluster herstellen.

Um eine Verbindung zu Ihrem Cluster herzustellen und Protokolldateien anzuzeigen, verwenden Sie AWS CLI

1. Stellen Sie mit dem folgenden Befehl eine SSH-Verbindung zu Ihrem Cluster her.
<mykeypair.key> Ersetzen Sie durch den vollständigen Pfad und Dateinamen Ihrer Schlüsselpaardatei. Beispiel, C:\Users\<username>\.ssh\mykeypair.pem.

```
aws emr ssh --cluster-id <j-2AL4XXXXXX5T9> --key-pair-file <~/mykeypair.key>
```

2. Navigieren Sie zu /mnt/var/log/spark, um auf die Spark-Protokolle auf dem Hauptknoten Ihres Clusters zuzugreifen. Sehen Sie sich dann die Dateien an diesem Speicherort an. Eine

Liste zusätzlicher Protokolldateien auf dem Hauptknoten finden Sie unter [Protokolldateien auf dem Primärknoten anzeigen](#).

```
cd /mnt/var/log/spark
ls
```

Verwenden Sie Amazon SageMaker AI Unified Studio, um Ihren Amazon EMR-Cluster zu verwalten

Amazon EMR on EC2 ist auch ein unterstützter Berechnungstyp für Amazon SageMaker AI Unified Studio. Informationen zur Verwendung und [Verwaltung von EMR auf EC2 EC2 Ressourcen in Amazon SageMaker AI Unified Studio](#) finden Sie unter [Amazon EMR verwalten](#) unter.

Schritt 3: Ihre Amazon-EMR-Ressourcen bereinigen

So beenden Sie Ihren Cluster

Nachdem Sie Arbeit an Ihren Cluster übermittelt und die Ergebnisse Ihrer PySpark Bewerbung eingesehen haben, können Sie den Cluster beenden. Durch das Beenden eines Clusters werden alle mit dem Cluster verbundenen Amazon EMR-Gebühren und EC2 Amazon-Instances gestoppt.

Wenn Sie einen Cluster kündigen, speichert Amazon EMR Metadaten über den Cluster zwei Monate lang kostenlos. Archivierte Metadaten helfen Ihnen dabei, [den Cluster für einen neuen Auftrag zu klonen](#) oder die Cluster-Konfiguration zu Referenzzwecken wiederaufzugreifen. Zu den Metadaten gehören keine Daten, die der Cluster in S3 schreibt, oder Daten, die in HDFS auf dem Cluster gespeichert sind.

Note

Mit der Amazon-EMR-Konsole können Sie nach dem Beenden des Clusters keinen Cluster aus der Listenansicht löschen. Ein beendeter Cluster verschwindet von der Konsole, wenn Amazon EMR seine Metadaten löscht.

Console

Um den Cluster mit der Konsole zu beenden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie Clusters und dann den Cluster aus, den Sie beenden möchten.
3. Wählen Sie im Dropdownmenü Aktionen die Option Cluster beenden aus.
4. Wählen Sie im Dialogfenster Beenden. Je nach Clusterkonfiguration kann die Kündigung 5 bis 10 Minuten dauern. Weitere Informationen zum Amazon-EMR-Cluster finden Sie unter [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#).

CLI

Um den Cluster zu beenden mit dem AWS CLI

1. Initiieren Sie den Vorgang zur Clusterbeendigung mit dem folgenden Befehl.
`<myClusterId>` Ersetzen Sie durch die ID Ihres Beispiel-Clusters. Der Befehl gibt keine Ausgabe zurück.

```
aws emr terminate-clusters --cluster-ids <myClusterId>
```

2. Um zu überprüfen, ob der Clusterbeendigungsprozess im Gange ist, überprüfen Sie den Clusterstatus mit dem folgenden Befehl.

```
aws emr describe-cluster --cluster-id <myClusterId>
```

Im Folgenden finden Sie eine Beispielausgabe im JSON-Format. Der Cluster Status sollte sich von **TERMINATING** zu **TERMINATED** ändern. Die Beendigung kann je nach Clusterkonfiguration 5 bis 10 Minuten dauern. Weitere Informationen zum Beenden eines Amazon-EMR-Clusters finden Sie unter [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#).

```
{
  "Cluster": {
    "Id": "j-xxxxxxxxxxxxx",
    "Name": "My Cluster Name",
    "Status": {
```

```
        "State": "TERMINATED",
        "StateChangeReason": {
            "Code": "USER_REQUEST",
            "Message": "Terminated by user request"
        }
    }
}
```

Löschen von S3-Ressourcen

Um zusätzliche Gebühren zu vermeiden, sollten Sie Ihren Amazon-S3-Bucket löschen. Durch das Löschen des Buckets werden alle Amazon-S3-Ressourcen für dieses Tutorial entfernt. Ihr Bucket sollte Folgendes enthalten:

- Das PySpark Skript
- Der Eingabedatensatz
- Ihr Ordner mit den Ausgabeergebnissen
- Ihr Ordner für Protokolldateien

Möglicherweise müssen Sie zusätzliche Schritte unternehmen, um gespeicherte Dateien zu löschen, wenn Sie PySpark das Skript oder die Ausgabe an einem anderen Ort gespeichert haben.

Note

Ihr Cluster muss beendet werden, bevor Sie Ihren Bucket löschen können. Andernfalls dürfen Sie den Bucket möglicherweise nicht leeren.

Um Ihren Bucket zu löschen, befolgen Sie die Anweisungen unter [Wie lösche ich einen S3-Bucket?](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Nächste Schritte

Sie haben jetzt Ihren ersten Amazon-EMR-Cluster von Anfang bis Ende gestartet. Sie haben auch wichtige EMR-Aufgaben wie das Vorbereiten und Einreichen von Big-Data-Anwendungen, das Anzeigen von Ergebnissen und das Beenden eines Clusters erledigt.

Verwenden Sie die folgenden Themen, um mehr über zu erfahren, wie Sie Ihren Amazon-EMR-Workflow anpassen können.

Erkunden Sie Big-Data-Anwendungen für Amazon EMR

Entdecken und vergleichen Sie die Big-Data-Anwendungen, die Sie auf einem Cluster installieren können, im [Amazon-EMR-Versionshandbuch](#). Der Versionshandbuch beschreibt jede EMR-Version und enthält Tipps zur Verwendung von Frameworks wie Spark und Hadoop auf Amazon EMR.

Planen Sie Cluster-Hardware, Netzwerke und Sicherheit

In diesem Tutorial haben Sie einen einfachen EMR-Cluster erstellt, ohne erweiterte Optionen zu konfigurieren. Mit den erweiterten Optionen können Sie EC2 Amazon-Instance-Typen, Cluster-Netzwerke und Cluster-Sicherheit angeben. Weitere Informationen zur Planung und Einführung eines Clusters, der Ihren Anforderungen entspricht, finden Sie unter [Planung, Konfiguration und Start von Amazon EMR-Clustern](#) und [Sicherheit in Amazon EMR](#).

Verwalten von Clustern

Erfahren Sie mehr über die Arbeit mit laufenden Clustern unter [Amazon EMR-Cluster verwalten](#). Um einen Cluster zu verwalten, können Sie eine Verbindung zum Cluster herstellen, Schritte debuggen und die Clusteraktivitäten und den Zustand verfolgen. Mit [EMR-verwalteter Skalierung](#) können Sie die Clusterressourcen auch an die Workload-Anforderungen anpassen.

Verwenden Sie eine andere Schnittstelle

Zusätzlich zur Amazon EMR-Konsole können Sie Amazon EMR mithilfe der AWS Command Line Interface, der Webservice-API oder einer der vielen unterstützten verwalten. AWS SDKs Weitere Informationen finden Sie unter [Verwaltungsschnittstellen](#).

Sie können auch auf vielfältige Weise mit Anwendungen interagieren, die auf Amazon-EMR-Clustern installiert sind. Einige Anwendungen wie Apache Hadoop veröffentlichen Weboberflächen, die Sie sich ansehen können. Weitere Informationen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Stöbern Sie im technischen Blog von EMR

Exemplarische Vorgehensweisen und ausführliche technische Diskussionen zu den neuen Amazon-EMR-Features finden Sie im [AWS -Big-Data-Blog](#).

Verwaltung von Amazon EMR-Clustern mit der Konsole

Die Konsole bietet eine aktualisierte Oberfläche, die Ihnen eine intuitive Möglichkeit bietet, Ihre Amazon EMR-Umgebung zu verwalten, und bietet Ihnen bequemen Zugriff auf Dokumentation, Produktinformationen und andere Ressourcen.

Funktionen der Konsole

Die Amazon EMR-Konsole ist unter der folgenden URL verfügbar:

- [Konsolen-URL — /emr https://console.aws.amazon.com](https://console.aws.amazon.com/emr)

In der folgenden Tabelle ist der Status der wichtigsten Komponenten der Amazon EMR-Konsole aufgeführt.

Komponente für Amazon-EMR-Konsole	Konsole	
EMR Studio	✓	
Cluster erstellen und verwalten	✓	
Blockieren des öffentlichen Zugriffs	✓	
Überwachen Sie CloudWatch Amazon-Ereignisse	✓	
Sicherheitskonfigurationen	✓	
Virtuelle Cluster (Amazon EMR in EKS)	✓	
Ihre Amazon Virtual Private Cloud-Subnetze anzeigen und verwalten 1	✓	
Notizbücher 2	✓	

¹ In der Konsole können Sie Ihre Amazon VPC-Subnetze im Bereich Netzwerk anzeigen und verwalten, wenn Sie einen Cluster erstellen.

² EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Zusammenfassung der Unterschiede

In diesem Abschnitt werden die Funktionen der Amazon EMR-Konsole beschrieben. Diese Funktionen lassen sich in die folgenden Kategorien einteilen:

- [Cluster-Kompatibilität in der Konsole](#)
- [Cluster erstellen](#)
- [Cluster-Details anzeigen oder bearbeiten](#)
- [Cluster anzeigen und nach ihnen suchen](#)
- [Unterschiede bei der Arbeit mit Sicherheitskonfigurationen](#)

Cluster-Kompatibilität in der Konsole

In einigen Fällen ist ein von Ihnen erstellter Cluster möglicherweise nicht mit der Konsole kompatibel. In der folgenden Liste werden die Kompatibilitätsanforderungen für die Amazon EMR-Konsole beschrieben.

- Die Konsole unterstützt Cluster, die in den Amazon EMR-Versionen 5.20.1 und höher erstellt wurden.
- Sie können Cluster mit automatischer Skalierung in der Konsole klonen, aber Sie können nur neue Cluster erstellen, wenn Sie sie manuell skalieren oder verwaltete Skalierung verwenden möchten.

Um Cluster der Version 5.20.1 und früher zu erstellen und mit ihnen zu arbeiten, können Sie das AWS Command Line Interface (AWS CLI) oder das AWS SDK verwenden.

Cluster erstellen

Funktion	Konsole	
Terminologie: Amazon-EMR-Cluster-Knotentypen	Primär, Core, Aufgabe	
Von Amazon EMR unterstützte Versionen 1	Amazon-EMR-Version 5.20.1 und höher	
Schnelles Starten eines Clusters	Klicken Sie im Bereich Zusammenfassung auf die Schaltfläche Cluster erstellen. Ihr Clustername darf die Zeichen <, >, \$, oder ` (Backtick) nicht enthalten.	
Konfiguration eines Timeouts für die Spot-Bereitstellung	Definieren Sie einen Timeout-Zeitraum für die Bereitstellung von Instances für jede Flotte in Ihrem Cluster.	
Servicerollen und Rolle im EC2 Amazon-Instanzprofil	Die Konsole erstellt keine Standardrollen. Sie müssen Rollen mit der IAM-Konsole erstellen oder eine bereits erstellte IAM-Rolle auswählen	
Transparenz der Cluster	Von der Amazon-EMR-Konsole aus können Sie einen Cluster nicht für alle Benutzer sichtbar machen. Ihre IAM-Richtlinie bestimmt den Clusterzugriff	

Funktion	Konsole	
Netzwerk – konfigurieren Sie private Subnetze	Sie müssen Amazon-S3-Endpunkte und NAT-Gateways von ihren jeweiligen Amazon-S3- und Amazon-VPC-Konsolen aus konfigurieren.	
Konsistente Ansicht des EMR-Dateisystems (EMRFS CV)	Mit der Veröffentlichung von Amazon S3 Strong read-after-write Consistency am 1. Dezember 2020 müssen Sie EMRFS CV nicht mit Ihren EMR-Clustern verwenden	
Debugging	Sie können Aufträge mithilfe der Benutzeroberfläche der Anwendung auf der Cluster-Detailseite debuggen	

¹ Sie können in der Konsole keine Cluster mit Versionen vor Amazon EMR 5.20.1 erstellen oder bearbeiten, aber alle vorhandenen Cluster, die mit Versionen vor 5.20.1 erstellt wurden, funktionieren weiterhin. Verwenden Sie die API oder CLI, um Cluster mit Amazon EMR-Versionen vor 5.20.1 zu erstellen und zu bearbeiten. Sie können alle Cluster mit der Konsole anzeigen, aber Konsolen, die vor 5.20.1 erstellt wurden, sind möglicherweise nicht mit neueren Funktionen kompatibel.

Cluster anzeigen und nach ihnen suchen

In der folgenden Tabelle wird erläutert, wie Sie die Amazon EMR-Konsole verwenden können, um Cluster anzuzeigen und nach ihnen zu suchen.

Note

Wenn Sie einen Datenfilter auf die Cluster-Liste anwenden, wird die gesamte Datenbank abgefragt. Wenn Sie jedoch eine Textzeichenfolge in das Suchfeld eingeben, gilt die Suche nur für die Ergebnisse, die die Liste clientseitig geladen hat.

Funktion	Konsole	
Anzeigen von Cluster-Details	Sie können die Cluster-ID auswählen, um umfassende Clusterdetails wie Konfigurationsoptionen, persistente Anwendungen UIs und Protokolle anzuzeigen.	
Suchen nach Clustern	Verwenden Sie ein einziges Suchfeld, um Textsuchabfragen einzugeben und Datenfilter wie „Status = Beliebiger aktiver Status“ zu erstellen und anzuwenden.	
Nach ausgefallenen Clustern suchen	Um nach ausgefallenen Clustern zu suchen, wenden Sie den Filter Status = Mit Fehlern beendet an.	

Cluster-Details anzeigen oder bearbeiten

Funktion	Konsole	

Funktion	Konsole	
Anzeige der Instances in Ihren Instance-Gruppen und Instance-Flotten sowie der Optionen für Skalierung, Bereitstellung, Größenänderung und Kündigung	Instance-Optionen und -Details finden Sie auf der Registerkarte Instances. Sehen Sie sich die Kündigungsoptionen auf der Registerkarte „Eigenschaften“ an.	
App UIs, Protokolle und Konfigurationen anzeigen (Apache-Spark -Benutzeroberfläche, Spark-Verlaufsservice, Apache-Tez-Benutzeroberfläche, YARN-Zeitleistenserver)	Sehen Sie sich die Cluster-Konfigurationen auf der Registerkarte Konfigurationen an. Starten Sie eine persistente Live-Anwendungsoberfläche, um die Protokolle für eine Anwendung auf der Registerkarte „Anwendungen“ anzuzeigen.	
Exportieren eines Clusters nach CLI	Die Option ist in den Aktionsmenüs „Cluster-Detail“ und „Listenansicht“ als „Befehl zum Klonen eines Clusters anzeigen“ verfügbar	

Unterschiede bei der Arbeit mit Sicherheitskonfigurationen

Funktion	Konsole	
Klonen von Sicherheitskonfigurationen	✓	
Föderierte Verwaltung mit Trino und Apache Ranger	✓	

Funktion	Konsole	
Eine Runtime-Rolle verwenden, um Arbeit an einen Cluster zu übermitteln ¹	✓	
Zugriff auf EMR-Dateisystem (EMRFS)	Amazon S3 Access Points	
AWS Lake Formation Zugriffskontrollen	Laufzeit-Rollen	

¹ Um eine Rolle während der Schrittübermittlung zu übergeben, muss Ihr Cluster eine Sicherheitskonfiguration mit einer angehängten IAM-Berechtigungsrichtlinie verwenden, sodass ein Benutzer nur die genehmigten Rollen weitergeben kann und Ihre Aufträge auf Amazon-EMR-Ressourcen zugreifen können. Weitere Informationen finden Sie unter [Schritte für Laufzeit-Rollen für Amazon EMR](#).

Amazon EMR Studio

Amazon EMR Studio ist eine webbasierte integrierte Entwicklungsumgebung (IDE) für vollständig verwaltete Jupyter Notebooks, die auf Amazon-EMR-Clustern ausgeführt werden. Sie können ein EMR Studio für Ihr Team einrichten, um in R, Python, Scala und geschriebene Anwendungen zu entwickeln, zu visualisieren und zu debuggen. PySpark EMR Studio ist in AWS Identity and Access Management (IAM) und IAM Identity Center integriert, sodass sich Benutzer mit ihren Unternehmensanmeldedaten anmelden können.

Sie können ein EMR Studio kostenlos erstellen. Wenn Sie EMR Studio verwenden, fallen Gebühren für Amazon-S3-Speicher und Amazon-EMR-Cluster an. Highlights, weitere Produktdetails und Preise finden Sie auf der Serviceseite für [Amazon EMR Studio](#).

Hauptfeatures von EMR Studio

Amazon EMR Studio bietet die folgenden Features:

- Authentifizieren Sie Benutzer mit AWS Identity and Access Management (IAM) oder AWS IAM Identity Center mit oder ohne [Trusted Identity Propagation und Ihrem Enterprise Identity Provider](#).
- Greifen Sie bei Bedarf auf Amazon-EMR-Cluster zu und starten Sie sie, um Jupyter-Notebook-Aufträge auszuführen.
- Stellen Sie auf EKS-Clustern eine Verbindung zu Amazon EMR her, um Arbeit einzureichen, während der Auftrag ausgeführt wird.
- Erkunden und speichern Sie Beispiel-Notebooks. Weitere Informationen zu Beispielnotizbüchern finden Sie im [EMR Studio Notebook Examples GitHub Repository](#).
- Analysieren Sie Daten mit Python PySpark, Spark Scala, Spark R oder SparkSQL und installieren Sie benutzerdefinierte Kernel und Bibliotheken.
- Arbeiten Sie in Echtzeit mit anderen Benutzern in demselben Workspace zusammen. Weitere Informationen finden Sie unter [Workspace-Zusammenarbeit in EMR Studio konfigurieren](#).
- Verwenden Sie den EMR Studio SQL Explorer, um Ihren Datenkatalog zu durchsuchen, SQL-Abfragen auszuführen und Ergebnisse herunterzuladen, bevor Sie mit den Daten in einem Notebook arbeiten.
- Führen Sie parametrisierte Notebooks als Teil von geplanten Workflows mit einem Orchestrierungstool wie Apache Airflow oder Amazon Managed Workflows für Apache Airflow aus.

Weitere Informationen finden Sie unter [Orchestrieren von Analyseaufträgen auf EMR Notebooks mithilfe von MWAA](#) im AWS Big Data-Blog.

- Verknüpfen Sie Code-Repositorys wie und. GitHub BitBucket
- Verfolgen und debuggen Sie Jobs mit dem Spark History Server, der Tez-Benutzeroberfläche oder dem YARN-Timeline-Server.

EMR Studio ist HIPAA-fähig und nach HITRUST CSF und SOC 2 zertifiziert. Weitere Informationen zur HIPAA-Konformität für Dienste finden Sie unter. AWS <https://aws.amazon.com/compliance/hipaa-compliance/> Weitere Informationen zur HITRUST CSF-Konformität für AWS Dienste finden Sie unter. <https://aws.amazon.com/compliance/hitrust/>

EMR Studio ist ebenfalls FedRamp konform. Weitere Informationen zu Compliance-Programmen, mit denen Amazon EMR konform ist, finden Sie unter [Konformitätsvalidierung für Amazon EMR](#). Weitere Informationen zu zusätzlichen Compliance-Programmen für AWS Services finden Sie unter [AWS Services im Umfang nach](#) Compliance-Programmen.

Verlauf der Features von Amazon EMR Studio

In dieser Tabelle sind Aktualisierungen zur Funktion Amazon EMR Managed Scaling aufgeführt.

Datum der Veröffentlichung	Funktion
5. Januar 2024	Unterstützung für EMR Studio in AWS GovCloud (US-Ost) und AWS GovCloud (US-West) hinzugefügt.
26. November 2023	Unterstützung für die Verarbeitung vertrauenswürdiger Identität en für EMR Studio mit IAM-Identity-Center-Authentifizierung hinzugefügt.
26. Oktober 2023	Es wurde die Möglichkeit hinzugefügt, eine Serverless-EMR-Anwendung mit interaktiven Funktionen zu erstellen.
28. Februar 2023	AWS KMS Kundenverwaltete Schlüsselunterstützung für die Speicherung von Anwendungsprotokollen für serverlose EMR-Anwendungen hinzugefügt.
23. Februar 2023	Es wurde die Erstellung von IAM-Rollen mit einem Klick für die Serverless-EMR-Auftragsübermittlung hinzugefügt. ECR-Suche

Datum der Veröffentlichung	Funktion
	hinzugefügt, wenn Sie ein benutzerdefiniertes Image für EMR-Serverless-Anwendungen auswählen.
27. Januar 2023	Notebooks mit Headless-Ausführung können den Fortschritt jeder einzelnen Zellenausführung <code>%execute_notebook</code> auf magische Weise verfolgen.
23. Januar 2023	Persistente Anwendungen wurden für schnellere Startzeiten optimiert.

Wie Amazon EMR Studio funktioniert

Ein Amazon EMR Studio ist eine Amazon-EMR-Ressource, die Sie für ein Team von Benutzern erstellen. EMR Studio ist eine webbasierte, integrierte Entwicklungsumgebung für vollständig verwaltete Jupyter-Notebooks, die auf Amazon-EMR-Clustern ausgeführt werden. Benutzer melden sich mit Unternehmensanmeldeinformationen bei einem Studio an.

Jedes EMR Studio, das Sie erstellen, verwendet die folgenden AWS Ressourcen:

- Eine Amazon Virtual Private Cloud (VPC) mit Subnetzen – Benutzer führen Studio-Kernel und -Anwendungen auf Amazon EMR und Amazon EMR auf EKS-Clustern in der angegebenen VPC aus. Ein EMR Studio kann eine Verbindung zu jedem Cluster in den Subnetzen herstellen, die Sie beim Erstellen des Studios angeben.
- IAM-Rollen und Berechtigungsrichtlinien – Um Benutzerberechtigungen zu verwalten, erstellen Sie IAM-Berechtigungsrichtlinien, die Sie der IAM-Identität eines Benutzers oder einer Benutzerrolle zuordnen. EMR Studio verwendet auch eine IAM-Dienstrolle und Sicherheitsgruppen, um mit anderen Diensten zusammenzuarbeiten. AWS Weitere Informationen erhalten Sie unter [Zugriffskontrolle](#) und [Definieren Sie Sicherheitsgruppen zur Steuerung des Netzwerkverkehrs in EMR Studio](#).
- Sicherheitsgruppen – EMR Studio verwendet Sicherheitsgruppen, um einen sicheren Netzwerkkanal zwischen dem Studio und einem EMR-Cluster einzurichten.
- Ein Amazon-S3-Backup-Speicherort – EMR Studio speichert Notebookarbeiten an einem Amazon-S3-Speicherort.

In den folgenden Schritten wird beschrieben, wie Sie ein EMR Studio erstellen und verwalten:

1. Erstellen Sie in Ihrem Studio entweder AWS-Konto mit IAM- oder IAM Identity Center-Authentifizierung. Detaillierte Anweisungen finden Sie unter [Richten Sie ein EMR Studio ein](#).
2. Weisen Sie Ihrem Studio Benutzer und Gruppen zu. Verwenden Sie Berechtigungsrichtlinien, um detaillierte Berechtigungen für jeden Benutzer festzulegen. Weitere Informationen finden Sie im Thema [EMR-Studio-Benutzer zuweisen und verwalten](#).
3. Beginnen Sie mit der Überwachung von EMR Studio-Aktionen mit AWS CloudTrail Ereignissen. Weitere Informationen finden Sie unter [Amazon-EMR-Studio-Aktionen überwachen](#).
4. Bieten Sie Studio-Benutzern mehr Cluster-Optionen mit Cluster-Vorlagen und Amazon EMR in EKS-verwalteten Endpunkten.

Authentifizierung und Benutzeranmeldung

Amazon EMR Studio unterstützt zwei Authentifizierungsmodi: den IAM-Authentifizierungsmodus und den IAM-Identity-Center-Authentifizierungsmodus. Der IAM-Modus verwendet AWS Identity and Access Management (IAM), der IAM Identity Center-Modus verwendet AWS IAM Identity Center. Wenn Sie ein EMR Studio erstellen, wählen Sie den Authentifizierungsmodus für alle Benutzer dieses Studios.

IAM-Authentifizierungsmodus

Im IAM-Authentifizierungsmodus können Sie entweder die IAM-Authentifizierung oder den IAM-Verbund verwenden.

Mit der IAM-Authentifizierung können Sie IAM-Identitäten wie Benutzer, Gruppen und Rollen in IAM verwalten. Sie gewähren Benutzern Zugriff auf ein Studio mit IAM-Berechtigungsrichtlinien und [attributbasierter Zugriffskontrolle](#) (ABAC).

Mit dem IAM-Verbund können Sie Vertrauen zwischen einem externen Identitätsanbieter (IdP) aufbauen und AWS Benutzeridentitäten über Ihren IdP verwalten.

Authentifizierungsmodus von IAM Identity Center

Mit dem IAM-Identity-Center-Authentifizierungsmodus können Sie Benutzern Verbundzugriff auf ein EMR Studio gewähren. Sie können IAM Identity Center verwenden, um Benutzer und Gruppen aus Ihrem IAM-Identity-Center-Verzeichnis, Ihrem vorhandenen Unternehmensverzeichnis oder einem externen IdP wie Azure Active Directory (AD) zu authentifizieren. Sie verwalten dann Benutzer mit Ihrem Identitätsanbieter (IdP).

EMR Studio unterstützt die Verwendung der folgenden Identitätsanbieter für IAM Identity Center:

- AWS Managed Microsoft AD und selbstverwaltetes Active Directory — Weitere Informationen finden Sie unter [Connect Ihrem Microsoft AD-Verzeichnis](#) herstellen.
- SAML-basierte Anbieter – Eine vollständige Liste finden Sie unter [Unterstützte Identitätsanbieter](#).
- Das IAM Identity Center-Verzeichnis — Weitere Informationen finden Sie unter [Identitäten in IAM Identity Center verwalten und Trusted Identity Propagation zwischen Anwendungen im Benutzerhandbuch](#).AWS IAM Identity Center

Wie sich die Authentifizierung auf die Anmeldung und die Benutzerzuweisung auswirkt

Der Authentifizierungsmodus, den Sie für EMR Studio wählen, wirkt sich darauf aus, wie sich Benutzer bei einem Studio anmelden, wie Sie einen Benutzer einem Studio zuweisen und wie Sie Benutzer autorisieren (ihnen Berechtigungen erteilen), Aktionen wie das Erstellen neuer Amazon-EMR-Cluster auszuführen.

In der folgenden Tabelle sind die Anmeldemethoden für EMR Studio nach Authentifizierungsmodus zusammengefasst.

EMR-Studio-Anmeldeoptionen nach Authentifizierungsmodus

Authentifizierungsmodus	Anmelde-Methode	Beschreibung
- IAM (Authentifizierung und Verbund) - IAM Identity Center	EMR-Studio-URL	Benutzer melden sich mit der Studio-Zugriffs-URL bei einem Studio an. Beispiel, <code>https://xxxxxxxxxxxxxxxxxxxxx.emrstudio-prod.us-east-1.amazonaws.com</code>. Benutzer geben IAM-Anmeldeinformationen ein, wenn Sie die IAM-Authentifizierung verwenden. Wenn Sie IAM-Verbund oder IAM Identity Center verwenden, leitet EMR Studio Benutzer zur Eingabe der Anmeldeinformationen zur Anmelde-URL Ihres Identitätsanbieters weiter. Im Zusammenhang mit dem Identitätsverbund wird diese Anmeldeoption als vom Serviceanbieter (SP) initiierte Anmeldung bezeichnet.

Authentifizierungsmodus	Anmelde-Methode	Beschreibung
- IAM (Verbund) - IAM Identity Center	Identitätsanbieter-(IdP)-Portal	Benutzer melden sich beim Portal Ihres Identitätsanbieters an, z. B. beim Azure-Portal, und starten die Amazon-EMR-Konsole. Nach dem Start der Amazon-EMR-Konsole wählen Benutzer ein Studio aus der Studio-Liste aus und öffnen es. Sie können EMR Studio auch als SAML-Anwendung konfigurieren, sodass sich Benutzer über das Portal Ihres Identitätsanbieters bei einem bestimmten Studio anmelden können. Anweisungen finden Sie unter So konfigurieren Sie ein EMR Studio als SAML-Anwendung in Ihrem IdP-Portal. Im Zusammenhang mit dem Identitätsverbund wird diese Anmeldeoption als vom Identitätsanbieter (IdP) initiierte Anmeldung bezeichnet.
IAM (Authentifizierung)	AWS Management Console	Benutzer melden sich AWS Management Console mit IAM-Anmeldeinformationen an und öffnen ein Studio aus der Studios-Liste in der Amazon EMR-Konsole.

In der folgenden Tabelle werden die Benutzerzuweisung und Autorisierung für EMR Studio nach Authentifizierungsmodus beschrieben.

EMR Studio Benutzerzuweisung und Autorisierung im Authentifizierungsmodus

Authentifizierungsmodus	Benutzerzuweisung	Benutzer-Autorisierung
IAM (Authentifizierung und Verbund)	Zulassen der <code>CreateStudioPresignedUrl</code> Aktion in einer IAM-Berechtigungsrichtlinie, die an eine IAM-Identität	Definieren Sie IAM-Berechtigungsrichtlinien, die bestimmte EMR-Studio-Aktionen zulassen.

Authentifizierungsmodus	Benutzerzuweisung	Benutzer-Autorisierung
	(Benutzer, Gruppe oder Rolle) angefügt ist. Lassen Sie für Verbundbenutzer die <code>CreateStudioPresignedUrl</code>-Aktion in einem IAM in der Berechtigungsrichtlinie zu, die Sie für die IAM-Rolle konfigurieren, die Sie für den Verbund verwenden. Verwenden Sie die attributbasierte Zugriffskontrolle (ABAC), um das Studio oder die Studios anzugeben, auf die der Benutzer zugreifen kann. Detaillierte Anweisungen finden Sie unter Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu.	Hängen Sie für native Benutzer die IAM-Berechtigungsrichtlinie an eine IAM-Identität (Benutzer, Gruppe oder Rolle) an. Lassen Sie für Verbundbenutzer Studio-Aktionen in der Berechtigungsrichtlinie zu, die Sie für die IAM-Rolle konfigurieren, die Sie für den Verbund verwenden. Weitere Informationen finden Sie unter EMR Studio-Benutzerberechtigungen für Amazon EC2 oder Amazon EKS konfigurieren.

Authentifizierungsmodus	Benutzerzuweisung	Benutzer-Autorisierung
IAM Identity Center	Ordnen Sie bei Studios, bei denen <code>IdCUserAssignment</code> auf <code>REQUIRED</code> eingestellt wurde, Benutzer dem Studio mit einer bestimmten Sitzungsrichtlinie zu. Weitere Informationen finden Sie unter Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu. Bei Studios, bei denen <code>IdCUserAssignment</code> auf <code>OPTIONAL</code> eingestellt wurde, kann jeder Identity-Center-Benutzer oder jede Identity-Center-Gruppe auf das Studio zugreifen.	Optional: Definieren Sie IAM-Sitzungsrichtlinien, die bestimmte EMR-Studio-Aktionen zulassen. Ordnen Sie einem Benutzer eine Sitzungsrichtlinie zu, wenn Sie ihn einem Studio zuweisen. Weitere Informationen finden Sie unter Benutzerberechtigungen für den IAM-Identity-Center-Authentifizierungsmodus.

Zugriffskontrolle

In Amazon EMR Studio konfigurieren Sie die Benutzerautorisierung (Berechtigungen) mit identitätsbasierten AWS Identity and Access Management (IAM)-Richtlinien. In diesen Richtlinien geben Sie zulässige Aktionen und Ressourcen sowie die Bedingungen an, unter denen die Aktionen zulässig sind.

Benutzerberechtigungen für den IAM-Authentifizierungsmodus

Um Benutzerberechtigungen festzulegen, wenn Sie die IAM-Authentifizierung für EMR Studio verwenden, lassen Sie Aktionen zu, z. B. `elasticmapreduce:RunJobFlow` in einer IAM-Berechtigungsrichtlinie. Sie können eine oder mehrere zu verwendende Berechtigungsrichtlinien erstellen. Sie könnten beispielsweise eine grundlegende Richtlinie erstellen, die es einem Benutzer nicht erlaubt, neue Amazon-EMR-Cluster zu erstellen, und eine weitere Richtlinie, die die Clustererstellung zulässt. Eine Liste aller Studio-Aktionen finden Sie unter [AWS Identity and Access Management Berechtigungen für EMR Studio-Benutzer](#).

Benutzerberechtigungen für den IAM-Identity-Center-Authentifizierungsmodus

Wenn Sie die IAM-Identity-Center-Authentifizierung verwenden, erstellen Sie eine einzelne EMR-Studio-Benutzerrolle. Die Benutzerrolle ist eine dedizierte IAM-Rolle, die ein Studio annimmt, wenn sich ein Benutzer anmeldet.

Sie fügen der EMR-Studio-Benutzerrolle IAM-Sitzungsrichtlinien hinzu. Eine Sitzungsrichtlinie ist eine spezielle Art von IAM-Berechtigungsrichtlinie, die einschränkt, was ein Verbundbenutzer während einer Studio-Anmeldesitzung tun kann. Mit Sitzungsrichtlinien können Sie spezifische Berechtigungen für einen Benutzer oder eine Gruppe festlegen, ohne mehrere Benutzerrollen für EMR Studio erstellen zu müssen.

Wenn Sie einem Studio [Benutzer und Gruppen zuweisen](#), ordnen Sie diesem Benutzer oder dieser Gruppe eine Sitzungsrichtlinie zu, um detaillierte Berechtigungen anzuwenden. Sie können die Sitzungsrichtlinie eines Benutzers oder einer Gruppe auch jederzeit aktualisieren. Amazon EMR speichert jede Sitzungsrichtlinienzuweisung, die Sie erstellen.

Weitere allgemeine Informationen zu Sitzungs-Richtlinien finden Sie unter [Berechtigungen und Richtlinien](#) im AWS Identity and Access Management -Benutzerhandbuch.

Workspaces

Workspaces sind die wichtigsten Bausteine von Amazon EMR Studio. Um Notebooks zu organisieren, erstellen Benutzer einen oder mehrere Workspaces in einem Studio. Weitere Informationen finden Sie unter [Lernen Sie EMR Studio-Arbeitsbereiche kennen](#).

Ähnlich wie [Workspaces in JupyterLab](#) behält ein Workspace den Status der Arbeit am Notizbuch bei. Die Workspace-Benutzeroberfläche erweitert die [JupyterLab](#) Open-Source-Oberfläche jedoch um zusätzliche Tools, mit denen Sie EMR-Cluster erstellen und anhängen, Jobs ausführen, Beispiel-Notebooks durchsuchen und Git-Repositorys verknüpfen können.

Die folgende Liste enthält die wichtigsten Features von EMR Studio Workspaces:

- Die Sichtbarkeit von Workspace basiert auf Studio. Workspaces, die Sie in einem Studio erstellen, sind in anderen Studios nicht sichtbar.
- Standardmäßig wird ein Workspace geteilt und kann von allen Studio-Benutzern gesehen werden. Es kann jedoch jeweils nur ein Benutzer einen Workspace öffnen und darin arbeiten. Um gleichzeitig mit anderen Benutzern zu arbeiten, können Sie [Workspace-Zusammenarbeit in EMR Studio konfigurieren](#)

- Sie können gleichzeitig mit anderen Benutzern in einem Workspace zusammenarbeiten, wenn Sie die Workspace-Zusammenarbeit aktivieren. Weitere Informationen finden Sie unter [Workspace-Zusammenarbeit in EMR Studio konfigurieren](#).
- Notebooks in einem Workspace verwenden denselben EMR-Cluster, um Befehle auszuführen. Sie können einen Workspace an einen Amazon EMR-Cluster anhängen, der auf Amazon ausgeführt wird, EC2 oder an einen virtuellen Amazon EMR on EKS-Cluster und verwalteten Endpunkt.
- Workspaces können zu einer anderen Availability Zone wechseln, die Sie den Subnetzen eines Studios zuordnen. Sie können einen Workspace beenden und neu starten, um den Failover-Prozess einzuleiten. Wenn Sie einen Workspace neu starten, startet EMR Studio den Workspace in einer anderen Availability Zone in der VPC des Studios, wenn das Studio mit Zugriff auf mehrere Availability Zones konfiguriert ist. Wenn das Studio nur über eine Availability Zone verfügt, versucht EMR Studio, den Workspace in einem anderen Subnetz zu starten. Weitere Informationen finden Sie unter [Beheben von Workspace-Verbindungsproblemen](#).
- Ein Workspace kann eine Verbindung zu Clustern in allen Subnetzen herstellen, die einem Studio zugeordnet sind.

Weitere Informationen zum Erstellen und Konfigurieren von EMR Studio Workspaces finden Sie unter [Lernen Sie EMR Studio-Arbeitsbereiche kennen](#).

Notebook-Speicher in Amazon EMR Studio

Wenn Sie einen Workspace verwenden, speichert EMR Studio die Zellen in Notebookdateien automatisch in regelmäßigen Abständen an dem Amazon-S3-Speicherort, der mit Ihrem Studio verknüpft ist. Bei diesem Backup-Prozess bleibt die Arbeit zwischen den Sitzungen erhalten, sodass Sie später darauf zurückgreifen können, ohne Änderungen an ein Git-Repository zu übertragen. Weitere Informationen finden Sie unter [Workspace-Inhalte in EMR Studio speichern](#).

Wenn Sie eine Notebook-Datei aus einem Workspace löschen, löscht EMR Studio die Backup-Version für Sie aus Amazon S3. Wenn Sie jedoch einen Workspace löschen, ohne zuerst die zugehörigen Notebookdateien zu löschen, verbleiben die Notebookdateien in Amazon S3 und es fallen weiterhin Speichergebühren an. Weitere Informationen hierzu finden Sie unter [Löschen Sie einen Workspace und Notebook-Dateien in EMR Studio](#).

Funktionen, Anforderungen und Einschränkungen von EMR Studio

Dieses Thema enthält Punkte, die Sie bei der Arbeit mit Amazon EMR Studio berücksichtigen sollten, darunter Überlegungen zu Regionen und Tools, Cluster-Anforderungen und technischen Einschränkungen.

Überlegungen

Beachten Sie Folgendes, wenn Sie mit EMR Studio arbeiten:

- EMR Studio ist in den folgenden AWS-Regionen Versionen verfügbar:
 - USA Ost (Ohio): (us-east-2)
 - USA Ost (Nord-Virginia): (us-east-1)
 - USA West (Nordkalifornien) (us-west-1)
 - USA West (Oregon): (us-west-2)
 - Afrika (Kapstadt) (af-south-1)
 - Asien-Pazifik (Hongkong) (ap-east-1)
 - Asien-Pazifik (Jakarta) (ap-southeast-3) *
 - Asien-Pazifik (Melbourne) (ap-southeast-4) *
 - Asien-Pazifik (Mumbai): (ap-south-1)
 - Asien-Pazifik (Osaka) (ap-northeast-3) *
 - Asien-Pazifik (Seoul): (ap-northeast-2)
 - Asien-Pazifik (Singapur): (ap-southeast-1)
 - Asien-Pazifik (Sydney): (ap-southeast-2)
 - Asien-Pazifik (Tokyo) (ap-northeast-1)
 - Kanada (Zentral): (ca-central-1)
 - Europa (Frankfurt) (eu-central-1)
 - Europa (Irland) (eu-west-1)
 - Europa (London) (eu-west-2)
 - Europa (Mailand) (eu-south-1)
 - Europa (Paris) (eu-west-3)
 - Europa (Spanien) (eu-south-2)
 - Europa (Stockholm) (eu-north-1)

- Europa (Zürich) (eu-central-2) *
- Israel (Tel Aviv) il-central-1) *
- Naher Osten (VAE) (me-central-1) *
- Südamerika (São Paulo) (sa-east-1)
- AWS GovCloud (US-Ost) (-1) gov-us-east
- AWS GovCloud (US-West) (gov-us-west-1)

* Die Live-Spark-Benutzeroberfläche wird in diesen Regionen nicht unterstützt.

- Damit Benutzer neue auf Amazon ausgeführte EMR-Cluster EC2 für einen Workspace bereitstellen können, können Sie ein EMR Studio mit einer Reihe von Cluster-Vorlagen verknüpfen. Administratoren können Clustervorlagen mit Service Catalog definieren und wählen, ob ein Benutzer oder eine Gruppe innerhalb eines Studios auf die Clustervorlagen zugreifen kann oder keine Clustervorlagen.
- Verwenden Sie die Amazon EMR-Servicerolle, wenn Sie Zugriffsberechtigungen für Notizbuchdateien definieren AWS Secrets Manager, die in Amazon S3 gespeichert sind oder aus denen Geheimnisse gelesen werden. Sitzungsrichtlinien werden mit diesen Berechtigungen nicht unterstützt.
- Sie können mehrere EMR-Studios erstellen, um den Zugriff auf EMR-Cluster auf verschiedenen Ebenen zu steuern. VPCs
- Verwenden Sie die AWS CLI , um Amazon EMR auf EKS-Clustern einzurichten. Anschließend können Sie die Studio-Oberfläche verwenden, um Cluster an Workspaces mit einem verwalteten Endpunkt anzuhängen, um Notebook-Jobs auszuführen.
- Wenn Sie Trusted Identity Propagation mit Amazon EMR verwenden, gibt es weitere Überlegungen, die auch für EMR Studio gelten. Weitere Informationen finden Sie unter [Überlegungen und Einschränkungen für Amazon EMR mit Identity-Center-Integration](#).
- EMR Studio unterstützt die folgenden magischen Python-Befehle nicht:
 - %alias
 - %alias_magic
 - %automagic
 - %macro
 - %%js
 - %%javascript
 - Ändern von proxy_user mit %configure

- Ändern von `KERNEL_USERNAME` mit `%env` oder `%set_env`
- Amazon EMR auf EKS-Clustern unterstützt keine SparkMagic Befehle für EMR Studio.
- Um mehrzeilige Scala-Anweisungen in Notebookzellen zu schreiben, stellen Sie sicher, dass alle Zeilen bis auf die letzte mit einem Punkt enden. Im folgenden Beispiel wird die richtige Syntax für mehrzeilige Scala-Anweisungen verwendet.

```
val df = spark.sql("SELECT * from table_name").  
    filter("col1='value']").  
    limit(50)
```

- Um die Sicherheit der Anwendungen außerhalb der Konsole zu erhöhen, die Sie möglicherweise mit Amazon EMR verwenden, sind die Anwendungs-Hosting-Domains in der Public Suffix List (PSL) registriert. Zu diesen Hosting-Domains gehören beispielsweise die folgenden: `emrstudio-prod.us-east-1.amazonaws.com`, `emrnotebooks-prod.us-east-1.amazonaws.com`, `emrappui-prod.us-east-1.amazonaws.com`. Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host--`-Präfix zu verwenden, falls Sie jemals sensible Cookies im Standard-Domainnamen einrichten müssen. Diese Vorgehensweise hilft Ihnen dabei, Ihre Domain vor CSRF (Cross-Site Request Forgery Attempts, Anforderungsfälschung zwischen Websites)-Versuchen zu schützen. Weitere Informationen finden Sie hier: [Set-Cookie](#)Seite im Mozilla Developer Network.
- Amazon EMR Studio Workspaces und Persistent UI Endpoints verwenden FIPS 140-validierte kryptografische Module für encryption-in-transit, was eine einfachere Einführung des Service für regulierte Workloads ermöglicht. Weitere Informationen zu Endpunkten mit persistenten Benutzeroberflächen finden Sie unter [Benutzerschnittstellen für persistente Anwendungen in Amazon EMR anzeigen](#). Weitere Informationen zu Notebooks finden Sie in der [Übersicht über Amazon EMR Notebooks](#).

Bekannte Probleme

- Ein EMR Studio, das IAM Identity Center mit aktivierter Weitergabe vertrauenswürdiger Identitäten verwendet, kann nur EMR-Clustern zugeordnet werden, die auch vertrauenswürdige Identitätsverteilung verwenden.
- Stellen Sie sicher, dass Sie Proxy-Management-Tools deaktivieren, z. B. FoxyProxy or SwitchyOmega im Browser, bevor Sie ein Studio erstellen. Aktive Proxys können Fehler verursachen, wenn Sie Studio erstellen wählen, und zu einer Netzwerkfehler-Fehlermeldung führen.

- Kernel, die auf Amazon EMR in EKS-Clustern ausgeführt werden, können aufgrund von Timeout-Problemen nicht gestartet werden. Wenn beim Starten des Kernels ein Fehler oder ein Problem auftritt, schließen Sie die Notebook-Datei, fahren Sie den Kernel herunter und öffnen Sie die Notebook-Datei erneut.
- Der Kernel-Neustartvorgang funktioniert nicht wie erwartet, wenn Sie einen Cluster von Amazon EMR in EKS verwenden. Nachdem Sie Kernel neu starten ausgewählt haben, aktualisieren Sie den Workspace, damit der Neustart wirksam wird.
- Wenn ein Workspace nicht an einen Cluster angehängt ist, wird eine Fehlermeldung angezeigt, wenn ein Studio-Benutzer eine Notebook-Datei öffnet und versucht, einen Kernel auszuwählen. Sie können diese Fehlermeldung ignorieren, indem Sie OK wählen, aber Sie müssen den Workspace an einen Cluster anhängen und einen Kernel auswählen, bevor Sie Notebook-Code ausführen können.
- Wenn Sie Amazon EMR 6.2.0 mit einer [Sicherheitskonfiguration](#) verwenden, um die Clustersicherheit einzurichten, erscheint die Workspace-Oberfläche leer und funktioniert nicht wie erwartet. Wir empfehlen Ihnen, eine andere unterstützte Version von Amazon EMR zu verwenden, wenn Sie Datenverschlüsselung oder Amazon-S3-Autorisierung für EMRFS für einen Cluster konfigurieren möchten. EMR Studio funktioniert mit den Amazon-EMR-Versionen 5.32.0 (Amazon-EMR-5.x-Serie) und 6.2.0 (Amazon-EMR-6.x-Serie) und höher.
- Wenn Sie [Debuggen Sie Amazon EMR, das auf Amazon-Jobs ausgeführt wird EC2](#), funktionieren die Links zur Spark-Benutzeroberfläche auf dem Cluster möglicherweise nicht oder werden nicht angezeigt. Um die Links zu regenerieren, erstellen Sie eine neue Notebook-Zelle und führen Sie den %%info-Befehl aus.
- Jupyter Enterprise Gateway bereinigt in den folgenden Amazon-EMR-Release-Versionen keine inaktiven Kernel auf dem Primärknoten eines Clusters: 5.32.0, 5.33.0, 6.2.0 und 6.3.0. Kernel im Leerlauf verbrauchen Rechenressourcen und können dazu führen, dass Cluster mit langer Laufzeit ausfallen. Mit dem folgenden Beispielskript können Sie die Kernelbereinigung im Leerlauf für Jupyter Enterprise Gateway konfigurieren. Sie können [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#) oder das Skript als Schritt einreichen. Weitere Informationen finden Sie unter [Befehle und Skripts auf einem Amazon-EMR-Cluster ausführen](#).

```
#!/bin/bash
sudo tee -a /emr/notebook-env/conf/jupyter_enterprise_gateway_config.py << EOF
c.MappingKernelManager.cull_connected = True
c.MappingKernelManager.cull_idle_timeout = 10800
c.MappingKernelManager.cull_interval = 300
```

EOF

```
sudo systemctl daemon-reload
sudo systemctl restart jupyter_enterprise_gateway
```

- Wenn Sie eine automatische Terminierungsrichtlinie mit den Amazon-EMR-Versionen 5.32.0, 5.33.0, 6.2.0 oder 6.3.0 verwenden, markiert Amazon EMR einen Cluster als inaktiv und beendet den Cluster möglicherweise automatisch, auch wenn Sie einen aktiven Python3-Kernel haben. Das liegt daran, dass bei der Ausführung eines Python3-Kernels kein Spark-Job auf dem Cluster gesendet wird. Um die automatische Terminierung mit einem Python3-Kernel zu verwenden, empfehlen wir die Verwendung von Amazon-EMR-Version 6.4.0 oder höher. Weitere Informationen zum Auto-Beenden finden Sie unter [Verwenden einer automatischen Terminierungsrichtlinie für die Amazon EMR-Clusterbereinigung](#).
- Wenn Sie `%%display` einen Spark DataFrame in einer Tabelle anzeigen, werden sehr breite Tabellen möglicherweise gekürzt. Sie können mit der rechten Maustaste auf die Ausgabe klicken und Neue Ansicht für Ausgabe erstellen auswählen, um eine scrollbare Ansicht der Ausgabe zu erhalten.
- Wenn Sie einen Spark-basierten Kernel wie PySpark Spark oder SparkR starten, wird eine Spark-Sitzung gestartet, und wenn Sie eine Zelle in einem Notizbuch ausführen, werden Spark-Jobs in dieser Sitzung in die Warteschlange gestellt. Wenn Sie eine laufende Zelle unterbrechen, wird der Spark-Auftrag weiter ausgeführt. Um den Spark-Auftrag zu beenden, sollten Sie die Cluster-interne Spark-Benutzeroberfläche verwenden. Weitere Informationen zur Verbindung mit einer Spark-Benutzeroberfläche finden Sie unter [Debuggen von Anwendungen und Aufträgen mit EMR Studio](#).
- Die Verwendung von Amazon EMR Studio Workspaces als Root-Benutzer in einem AWS-Konto verursacht einen 403: Forbidden Fehler. Dies liegt daran, dass die Jupyter Enterprise Gateway-Konfiguration in Amazon EMR dem Root-Benutzer keinen Zugriff gewährt. Wir empfehlen, den Root-Benutzer nicht für Ihre täglichen Aufgaben zu verwenden. Weitere Authentifizierungsoptionen finden Sie unter [AWS Identity and Access Management Amazon EMR](#).

Feature-Einschränkungen

Amazon EMR Studio unterstützt die folgenden Amazon-EMR-Feature nicht:

- Anhängen und Ausführen von Aufträgen auf EMR-Clustern mit einer Sicherheitskonfiguration, die die Kerberos-Authentifizierung spezifiziert
- Cluster mit mehreren Primärknoten

- Cluster, die EC2 Amazon-Instances auf Basis von AWS Graviton2 für Amazon EMR 6.x-Versionen unter 6.9.0 und 5.x-Versionen unter 5.36.1 verwenden

Die folgenden Features werden von einem Studio, das die Verbreitung vertrauenswürdiger Identitäten verwendet, nicht unterstützt:

- Erstellen von EMR-Clustern ohne Vorlage.
- Verwenden von EMR-Serverless-Anwendungen.
- Starten von Amazon EMR in EKS-Clustern.
- Verwenden einer Laufzeitrolle.
- Aktivieren der Zusammenarbeit mit SQL Explorer oder Workspace.

Service-Limits für EMR Studio

In der folgenden Tabelle werden die Service-Limits für EMR Studio aufgeführt.

Item	Limit
EMR Studios	AWS Maximal 100 pro Konto
Subnetze	Maximal fünf für jedes EMR-Studio
IAM-Identity-Center-Gruppen	Maximal fünf für jedes EMR-Studio
Benutzer von IAM Identity Center	Maximal 100 für jedes EMR-Studio

Bewährte Methoden für VPC und Subnetze für EMR Studio

Verwenden Sie die folgenden bewährten Methoden, um eine Amazon Virtual Private Cloud (Amazon VPC) mit Subnetzen für EMR Studio einzurichten:

- Sie können in Ihrer VPC maximal fünf Subnetze angeben, die Sie dem Studio zuordnen möchten. Wir empfehlen, dass Sie mehrere Subnetze in verschiedenen Availability Zones bereitstellen, um die Workspace-Verfügbarkeit zu unterstützen und Studio-Benutzern Zugriff auf Cluster in verschiedenen Availability Zones zu gewähren. Weitere Informationen zum Arbeiten mit VPCs

Subnetzen und Availability Zones finden Sie unter [VPCs Subnetze](#) im Benutzerhandbuch.Amazon Virtual Private Cloud

- Die von Ihnen angegebenen Subnetze sollten miteinander kommunizieren können.
- Damit Benutzer einen Workspace mit öffentlich gehosteten Git-Repositorys verknüpfen können, sollten Sie nur private Subnetze angeben, die über Network Address Translation (NAT) Zugriff auf das Internet haben. Weitere Informationen zum Einrichten eines privaten Subnetzes für Amazon EMR finden Sie unter [Private Subnetze](#).
- Wenn Sie Amazon EMR auf EKS mit EMR Studio verwenden, muss mindestens ein gemeinsames Subnetz zwischen Ihrem Studio und dem Amazon-EKS-Cluster vorhanden sein, den Sie zum Registrieren eines virtuellen Clusters verwenden. Andernfalls wird Ihr verwalteter Endpunkt nicht als Option in Studio Workspaces angezeigt. Sie können einen Amazon-EKS-Cluster erstellen und ihn einem Subnetz zuordnen, das zum Studio gehört, oder Sie können ein Studio erstellen und die Subnetze Ihres EKS-Clusters angeben.
- Wenn Sie Amazon EMR in EKS mit EMR Studio verwenden möchten, wählen Sie die VPC für Ihre Amazon-EKS-Cluster-Worker-Knoten aus.

Anforderungen für Amazon EMR-Cluster

Amazon EMR-Cluster, die auf Amazon ausgeführt werden EC2

Alle Amazon EMR-Cluster, die auf Amazon ausgeführt werden und EC2 die Sie für einen EMR Studio Workspace erstellen, müssen die folgenden Anforderungen erfüllen. Cluster, die Sie mit der EMR-Studio-Oberfläche erstellen, erfüllen diese Anforderungen automatisch.

- Der Cluster muss Amazon-EMR-Versionen 5.32.0 (Amazon EMR 5.x-Serie) oder 6.2.0 (Amazon EMR 6.x-Serie) oder höher verwenden. Sie können mit der Amazon EMR-Konsole oder dem SDK einen Cluster erstellen und ihn dann an einen EMR Studio Workspace anhängen. AWS Command Line Interface Studio-Benutzer können auch Cluster bereitstellen und anhängen, wenn sie einen Amazon-EMR-Workspace erstellen oder darin arbeiten. Weitere Informationen finden Sie unter [Einen Computer an einen EMR Studio Workspace anhängen](#).
- Dieser Cluster muss sich innerhalb einer Amazon Virtual Private Cloud befinden. Die EC2 -Classic-Plattform wird nicht unterstützt.
- Auf dem Cluster müssen Spark, Livy und Jupyter Enterprise Gateway installiert sein. Wenn Sie den Cluster für SQL Explorer verwenden möchten, sollten Sie sowohl Presto als auch Spark installieren.

- Um SQL Explorer verwenden zu können, muss der Cluster Amazon-EMR-Version 5.34.0 oder höher oder Version 6.4.0 oder höher verwenden und Presto installiert sein. Wenn Sie den AWS Glue-Datenkatalog als Hive-Metastore für Presto angeben möchten, müssen Sie ihn auf dem Cluster konfigurieren. Weitere Informationen finden Sie unter [Verwendung von Presto mit dem AWS Glue Data Catalog](#).
- Der Cluster muss sich in einem privaten Subnetz mit Network Address Translation (NAT) befinden, um öffentlich gehostete Git-Repositorys mit EMR Studio verwenden zu können.

Wir empfehlen die folgenden Clusterkonfigurationen, wenn Sie mit EMR Studio arbeiten.

- Stellen Sie den Bereitstellungsmodus für Spark-Sitzungen auf den Clustermodus ein. Im Clustermodus werden die Anwendungsmasterprozesse auf den Core-Knoten und nicht auf dem Primärknoten eines Clusters platziert. Dadurch wird der Primärknoten von potenziellem Speicherdruck entlastet. Weitere Informationen finden Sie unter [Cluster Mode Overview](#) in der Apache Spark-Dokumentation.
- Ändern Sie das Livy-Timeout wie in der folgenden Beispielkonfiguration von der Standardeinstellung von einer Stunde auf sechs Stunden.

```
{
  "classification": "livy-conf",
  "Properties": {
    "livy.server.session.timeout": "6h",
    "livy.spark.deploy-mode": "cluster"
  }
}
```

- Erstellen Sie verschiedene Instance-Flotten mit bis zu 30 Instances und wählen Sie mehrere Instance-Typen in Ihrer Spot Instance-Flotte aus. Sie könnten beispielsweise die folgenden arbeitsspeicheroptimierten Instance-Typen für Spark-Workloads angeben: r5.2x, r5.4x, r5.8x, r5.12x, r5.16x, r4.2x, r4.4x, r4.8x, r4.12, usw. Weitere Informationen finden Sie unter [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#).
- Verwenden Sie die kapazitätsoptimierte Zuweisungsstrategie für Spot-Instances, um Amazon EMR dabei zu unterstützen, auf der Grundlage von Echtzeit-Kapazitätsinformationen von Amazon eine effektive Instance-Auswahl zu treffen. EC2 Weitere Informationen finden Sie unter [Zuweisungsstrategie für Flotten](#).
- Aktivieren Sie die verwaltete Skalierung in Ihrem Cluster. Legen Sie den Parameter für die maximale Anzahl an Core-Knoten auf die minimale persistente Kapazität fest, die Sie verwenden

möchten, und konfigurieren Sie die Skalierung für eine gut diversifizierte Task-Flotte, die auf Spot Instances ausgeführt wird, um Kosten zu sparen. Weitere Informationen finden Sie unter [Verwenden der verwalteten Skalierung in Amazon EMR](#).

Wir bitten Sie außerdem dringend, Amazon EMR Block Public Access aktiviert zu lassen und den eingehenden SSH-Verkehr auf vertrauenswürdige Quellen zu beschränken. Durch den eingehenden Zugriff auf einen Cluster können Benutzer Notebooks auf dem Cluster ausführen. Weitere Informationen erhalten Sie unter [Verwenden von Amazon EMR Block Public Access](#) und [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Cluster von Amazon EMR in EKS

Zusätzlich zu EMR-Clustern, die auf Amazon ausgeführt werden EC2, können Sie Amazon EMR on EKS-Cluster für EMR Studio mithilfe von einrichten und verwalten. AWS CLI Richten Sie Amazon EMR auf EKS-Clustern gemäß den folgenden Richtlinien ein:

- Erstellen Sie einen verwalteten HTTPS-Endpunkt für den Cluster von Amazon EMR in EKS. Benutzer hängen einen Workspace an einen verwalteten Endpunkt an. Der Amazon Elastic Kubernetes Service (EKS)-Cluster, den Sie zur Registrierung eines virtuellen Clusters verwenden, muss über ein privates Subnetz verfügen, um verwaltete Endgeräte zu unterstützen.
- Verwenden Sie einen Amazon-EKS-Cluster mit mindestens einem privaten Subnetz und Network Address Translation (NAT), wenn Sie öffentlich gehostete Git-Repositorys verwenden möchten.
- Vermeiden Sie die Verwendung von [Amazon EKS-optimiertem Arm Amazon Linux AMIs](#), das für Amazon EMR auf von EKS verwalteten Endpunkten nicht unterstützt wird.
- Vermeiden Sie die AWS Fargate ausschließliche Verwendung von Amazon EKS-Clustern, die nicht unterstützt werden.

Amazon EMR Studio konfigurieren

Dieser Abschnitt richtet sich an EMR-Studio-Administratoren. Es behandelt, wie Sie ein EMR Studio für Ihr Team einrichten, und enthält Anweisungen für Aufgaben wie das Zuweisen von Benutzern und Gruppen, das Einrichten von Clustervorlagen und das Optimieren von Apache Spark für EMR Studio.

Themen

- [Administratorberechtigungen zum Erstellen und Verwalten eines EMR Studios](#)
- [Richten Sie ein EMR Studio ein](#)

- [Amazon EMR Studio-Ressourcen überwachen, aktualisieren und löschen](#)
- [Verschlüsselung von EMR Studio Workspace-Notizbüchern und -Dateien](#)
- [Definieren Sie Sicherheitsgruppen zur Steuerung des Netzwerkverkehrs in EMR Studio](#)
- [AWS CloudFormation Vorlagen für Amazon EMR Studio erstellen](#)
- [Zugriff und Berechtigungen für Git-basierte Repositorys einrichten](#)
- [Spark-Aufträge in EMR Studio optimieren](#)

Administratorberechtigungen zum Erstellen und Verwalten eines EMR Studios

Die auf dieser Seite beschriebenen IAM-Berechtigungen ermöglichen es Ihnen, ein EMR Studio zu erstellen und zu verwalten. Weitere Informationen zu erforderlichen Berechtigungen finden Sie unter [Erforderliche Berechtigungen zum Verwalten eines EMR-Studios](#).

Erforderliche Berechtigungen zum Verwalten eines EMR-Studios

In der folgenden Tabelle sind die Vorgänge im Zusammenhang mit der Erstellung und Verwaltung eines EMR-Studios aufgeführt. In der Tabelle werden auch die für jeden Vorgang erforderlichen Berechtigungen angezeigt.

Note

Sie benötigen nur IAM-Identity-Center- und SessionMapping-Studio-Aktionen, wenn Sie den IAM-Identity-Center-Authentifizierungsmodus verwenden.

Berechtigungen zum Erstellen und Verwalten eines EMR Studios

Operation	Berechtigungen
Ein Studio erstellen	<pre>"elasticmapreduce:CreateStudio", "sso:CreateApplication", "sso:PutApplicationAuthentic ationMethod", "sso:PutApplicationGrant", "sso:PutApplicationAccessScope", "sso:PutApplicationAssignmentConfi guration",</pre>

Operation	Berechtigungen
	<code>"iam:PassRole"</code>
Ein Studio beschreiben	<code>"elasticmapreduce:DescribeStudio", "sso:GetManagedApplicationInstance"</code>
Studios auflisten	<code>"elasticmapreduce:ListStudios"</code>
Ein Studio löschen	<code>"elasticmapreduce:DeleteStudio", "sso:DeleteApplication", "sso:DeleteApplicationAuthenticati onMethod", "sso:DeleteApplicationAccessScope", "sso:DeleteApplicationGrant"</code>

Additional permissions required when you use IAM Identity Center mode

Einem Studio Benutzer oder Gruppen zuweisen	<code>"elasticmapreduce:CreateStudioSessio nMapping", "sso:GetProfile", "sso:ListDirectoryAssociations", "sso:ListProfiles", "sso:AssociateProfile", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListInstances", "sso:CreateApplicationAssignment", "sso:DescribeInstance", "organizations:DescribeOrga nization", "organizations:ListDelegatedAdmini strators", "sso:CreateInstance", "sso:DescribeRegisteredRegions", "sso:GetSharedSsoConfiguration", "iam:ListPolicies"</code>

Operation	Berechtigungen
Rufen Sie die Studio-Zuweisungsdetails für einen bestimmten Benutzer oder eine bestimmte Gruppe ab	<pre>"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:DescribeApplication", "elasticmapreduce:GetStudioSessionMapping"</pre>
Alle Benutzer und Gruppen auflisten, die einem Studio zugewiesen sind	<pre>"elasticmapreduce:ListStudioSessionMappings"</pre>
Aktualisieren Sie die Sitzungsrichtlinie, die einem Benutzer oder einer Gruppe zugewiesen ist, die einem Studio zugewiesen ist	<pre>"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:DescribeApplication", "sso:DescribeInstance", "elasticmapreduce:UpdateStudioSessionMapping"</pre>
Einen Benutzer oder eine Gruppe aus einem Studio entfernen	<pre>"elasticmapreduce>DeleteStudioSessionMapping", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListDirectoryAssociations", "sso:GetProfile", "sso:DescribeApplication", "sso:DescribeInstance", "sso:ListProfiles", "sso:DisassociateProfile", "sso>DeleteApplicationAssignment", "sso:ListApplicationAssignments"</pre>

So erstellen Sie eine Richtlinie mit Administratorberechtigungen für EMR Studio

1. Folgen Sie den Anweisungen unter [IAM-Richtlinien erstellen](#), um eine Richtlinie anhand eines der folgenden Beispiele zu erstellen. Welche Berechtigungen Sie benötigen, hängt von Ihrem [Authentifizierungsmodus für EMR Studio](#) ab.

Fügen Sie Ihre eigenen Werte für diese Elemente ein:

- Ersetzen Sie `<your-resource-ARN>` , um den Amazon-Ressourcennamen (ARN) des Objekts oder der Objekte anzugeben, die die Anweisung für Ihre Anwendungsfälle abdeckt.
- `<region>` Ersetzen Sie es durch den Code des AWS-Region Ortes, an dem Sie das Studio erstellen möchten.
- `<aws-account-id>` Ersetzen Sie es durch die ID des AWS Kontos für das Studio.
- Ersetzen Sie `<EMRStudio-Service-Role>` und `<EMRStudio-User-Role>` durch die Namen Ihrer [EMR Studio-Service-rolle](#) und [EMR Studio-Benutzerrolle](#).

Example Beispielrichtlinie: Administratorberechtigungen, wenn Sie den IAM-Authentifizierungsmodus verwenden

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/*",
      "Action": [
        "elasticmapreduce:CreateStudio",
        "elasticmapreduce:DescribeStudio",
        "elasticmapreduce>DeleteStudio"
      ]
    },
    {
      "Effect": "Allow",
      "Resource": "<your-resource-ARN>",
      "Action": [
        "elasticmapreduce:ListStudios"
      ]
    }
  ]
}
```

```

        "Effect": "Allow",
        "Resource": [
            "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>"
        ],
        "Action": "iam:PassRole"
    }
]
}

```

Example Beispielrichtlinie: Administratorberechtigungen, wenn Sie den IAM-Identity-Center-Authentifizierungsmodus verwenden

Note

Identity Center und das Identity Center-Verzeichnis unterstützen die Angabe eines ARN im Ressourcenelement einer IAM-Richtlinienanweisung APIs nicht. Um den Zugriff auf IAM Identity Center und IAM Identity Center Directory zu ermöglichen, spezifizieren die folgenden Berechtigungen alle Ressourcen, „Resource“: „*“ für IAM-Identity-Center-Aktionen. Weitere Informationen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für IAM Identity Center Directory](#).

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce>DeleteStudio",
                "elasticmapreduce:CreateStudioSessionMapping",
                "elasticmapreduce:GetStudioSessionMapping",
                "elasticmapreduce:UpdateStudioSessionMapping",
                "elasticmapreduce>DeleteStudioSessionMapping"
            ]
        },
        {
            "Effect": "Allow",

```



```

    "Resource": "<your-resource-ARN>",
    "Action": [
        "elasticmapreduce:ListStudios",
        "elasticmapreduce:ListStudioSessionMappings"
    ]
},
{
    "Effect": "Allow",
    "Resource": [
        "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>",
        "arn:aws:iam::<aws-account-id>:role/<EMRStudio-User-Role>"
    ],
    "Action": "iam:PassRole"
},
{
    "Effect": "Allow",
    "Resource": "*",
    "Action": [
        "sso:CreateApplication",
        "sso:PutApplicationAuthenticationMethod",
        "sso:PutApplicationGrant",
        "sso:PutApplicationAccessScope",
        "sso:PutApplicationAssignmentConfiguration",
        "sso:DescribeApplication",
        "sso:DeleteApplication",
        "sso:DeleteApplicationAuthenticationMethod",
        "sso:DeleteApplicationAccessScope",
        "sso:DeleteApplicationGrant",
        "sso:ListInstances",
        "sso:CreateApplicationAssignment",
        "sso:DeleteApplicationAssignment",
        "sso:ListApplicationAssignments",
        "sso:DescribeInstance",
        "sso:AssociateProfile",
        "sso:DisassociateProfile",
        "sso:GetProfile",
        "sso:ListDirectoryAssociations",
        "sso:ListProfiles",
        "sso-directory:SearchUsers",
        "sso-directory:SearchGroups",
        "sso-directory:DescribeUser",
        "sso-directory:DescribeGroup",
        "organizations:DescribeOrganization",
        "organizations:ListDelegatedAdministrators",
    ]
}

```

```
        "sso:CreateInstance",
        "sso:DescribeRegisteredRegions",
        "sso:GetSharedSsoConfiguration",
        "iam:ListPolicies"
    ]
}
]
```

2. Hängen Sie die Richtlinie an Ihre IAM-Identität (Benutzer, Rolle oder Gruppe) an. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Richten Sie ein EMR Studio ein

Gehen Sie wie folgt vor, um ein EMR Studio einzurichten.

Bevor Sie beginnen

Note

Wenn Sie planen, EMR Studio mit Amazon EMR in EKS zu verwenden, empfehlen wir, zuerst Amazon EMR in EKS für EMR Studio einzurichten, bevor Sie ein Studio einrichten.

Bevor Sie ein EMR-Studio einrichten, stellen Sie sicher, dass Sie über die folgenden Elemente verfügen:

- Ein AWS-Konto. Detaillierte Anweisungen finden Sie unter [Bevor Sie Amazon EMR einrichten](#).
- Berechtigungen zum Erstellen und Verwalten eines EMR Studios. Weitere Informationen finden Sie unter [the section called “Administratorberechtigungen zum Erstellen eines EMR-Studios”](#).
- Ein Amazon-S3-Bucket, in dem EMR Studio die Workspaces und Notebookdateien in Ihrem Studio sichern kann. Anweisungen finden Sie unter [Erstellen eines Buckets](#) im Amazon Simple Storage Service (S3)-Benutzerhandbuch.
- Wenn Sie eine Verbindung zu einem Amazon EMR on EC2 - oder Amazon EMR on EKS-Cluster herstellen oder Git-Repositorys verwenden möchten, benötigen Sie eine Amazon Virtual Private Cloud (VPC) für das Studio und maximal fünf Subnetze. Sie benötigen keine VPC, um EMR Studio mit EMR Serverless zu verwenden. Tipps zur Netzwerkkonfiguration finden Sie unter [Bewährte Methoden für VPC und Subnetze für EMR Studio](#).

So richten Sie ein EMR Studio ein

1. [Einen Authentifizierungsmodus für Amazon EMR Studio auswählen](#)
2. Erstellen Sie die folgenden Studio-Ressourcen.
 - [Eine EMR-Studio-Servicerolle erstellen](#)
 - [EMR Studio-Benutzerberechtigungen für Amazon EC2 oder Amazon EKS konfigurieren](#)
 - (Optional) [Definieren Sie Sicherheitsgruppen zur Steuerung des Netzwerkverkehrs in EMR Studio.](#)
3. [Ein EMR Studio erstellen](#)
4. [Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu](#)

Nachdem Sie diese Einrichtungs-Schritte abgeschlossen haben, fahren Sie mit [Ein Amazon EMR Studio verwenden](#) fort.

Einen Authentifizierungsmodus für Amazon EMR Studio auswählen

EMR Studio unterstützt zwei Authentifizierungsmodi: den IAM-Authentifizierungsmodus und den IAM-Identity-Center-Authentifizierungsmodus. Der IAM-Modus verwendet AWS Identity and Access Management (IAM), während der IAM Identity Center-Modus verwendet wird. AWS IAM Identity Center Wenn Sie ein EMR Studio erstellen, wählen Sie den Authentifizierungsmodus für alle Benutzer dieses Studios. Weitere Informationen zu diesen verschiedenen Authentifizierungsmodi finden Sie unter [Authentifizierung und Benutzeranmeldung](#).

Verwenden Sie die folgende Tabelle, um einen Authentifizierungsmodus für EMR Studio auszuwählen.

Wenn Sie ...	Wir empfehlen...
Bereits mit der IAM-Authentifizierung oder dem IAM-Verbund vertraut sind oder diese eingerichtet haben	IAM-Authentifizierungsmodus , welches die folgenden Vorteile bietet: • Ermöglicht ein Quick Setup für EMR Studio, wenn Sie bereits Identitäten wie Benutzer und Gruppen in IAM verwalten. • Funktioniert mit Identitätsanbietern, die mit OpenID Connect (OIDC) oder Security

Wenn Sie ...	Wir empfehlen...
	Assertion Markup Language 2.0 (SAML 2.0) kompatibel sind. • Unterstützt die Verwendung mehrerer Identitätsanbieter mit demselben AWS-Konto • Verfügbar in einer Vielzahl von AWS-Regionen • Konform mit SOC 2.
Neu bei AWS Amazon EMR	Authentifizierungsmodus von IAM Identity Center, welches die folgenden Features bietet: • Unterstützt die einfache Zuweisung von AWS Ressourcen durch Benutzer und Gruppen. • Funktioniert mit Microsoft-Active-Directory- und SAML-2.0-Identitätsanbietern. • Erleichtert die Einrichtung eines Verbunds mit mehreren Konten, sodass Sie den Verbund nicht für jedes Konto AWS-Konto in Ihrer Organisation separat konfigurieren müssen.

IAM-Authentifizierungsmodus für Amazon EMR Studio einrichten

Im IAM-Authentifizierungsmodus können Sie entweder die IAM-Authentifizierung oder den IAM-Verbund verwenden. Mit der IAM-Authentifizierung können Sie IAM-Identitäten wie Benutzer, Gruppen und Rollen in IAM verwalten. Sie gewähren Benutzern Zugriff auf ein Studio mit IAM-Berechtigungsrichtlinien und [attributbasierter Zugriffskontrolle](#) (ABAC). Mit dem IAM-Verbund können Sie Vertrauen zwischen einem externen Identitätsanbieter (IdP) aufbauen und AWS Benutzeridentitäten über Ihren IdP verwalten.

Note

Wenn Sie IAM bereits verwenden, um den Zugriff auf AWS Ressourcen zu steuern, oder wenn Sie Ihren Identity Provider (IdP) bereits für IAM konfiguriert haben, finden Sie weitere

Informationen unter [Benutzerberechtigungen für den IAM-Authentifizierungsmodus](#) So legen Sie Benutzerberechtigungen fest, wenn Sie den IAM-Authentifizierungsmodus für EMR Studio verwenden.

Den IAM-Verbund für Amazon EMR Studio verwenden

Um den IAM-Verbund für EMR Studio zu verwenden, erstellen Sie eine Vertrauensbeziehung zwischen Ihrem AWS-Konto und Ihrem Identitätsanbieter (IdP) und ermöglichen Verbundbenutzern den Zugriff auf AWS Management Console. Die Schritte, die Sie zum Aufbau dieser Vertrauensbeziehung ergreifen, hängen vom Verbundstandard Ihres IdP ab.

Im Allgemeinen führen Sie die folgenden Aufgaben aus, um den Verbund mit einem externen IdP zu konfigurieren. Vollständige Anweisungen finden Sie unter [Aktivieren des Zugriffs von SAML-2.0-Verbundbenutzern auf AWS Management Console](#) und [Aktivieren des benutzerdefinierten Identity Broker-Zugriffs auf AWS Management Console](#) im AWS Identity and Access Management - Benutzerhandbuch.

1. Sammeln Sie Informationen von Ihrem IdP. Dies bedeutet in der Regel die Generierung eines Metadatendokuments zur Validierung von SAML-Authentifizierungsanfragen von Ihrem IdP.
2. Erstellen Sie eine IAM-Entität eines Identitätsanbieters, um Informationen über Ihren IdP zu speichern. Anweisungen finden Sie unter [IAM-Identitätsanbieter erstellen](#).
3. Erstellen Sie eine oder mehrere IAM-Rollen für Ihren IdP. EMR Studio weist einem Verbundbenutzer eine Rolle zu, wenn dieser sich anmeldet. Die Rolle ermöglicht es dem Identitätsanbieter, temporäre Sicherheitsanmeldeinformationen für den Zugriff auf AWS anzufordern. Anweisungen finden Sie unter [Erstellen einer Rolle für einen Drittanbieter-Identitätsanbieter \(Verbund\)](#). Die Berechtigungsrichtlinien, die Sie der Rolle zuweisen, bestimmen, was Verbundbenutzer in AWS und in einem EMR Studio tun können. Weitere Informationen finden Sie unter [Benutzerberechtigungen für den IAM-Authentifizierungsmodus](#).
4. (Für SAML-Anbieter) Vervollständigen Sie die SAML-Vertrauensstellung, indem Sie Ihren IdP mit Informationen zu AWS und den Rollen konfigurieren, die Verbundbenutzer annehmen sollen. Dieser Konfigurationsprozess schafft Vertrauen zwischen Ihrem IdP und AWS. Weitere Informationen finden Sie unter [Konfigurieren des SAML-2.0-Identitätsanbieters mit Vertrauensstellung für die vertrauende Seite und Hinzufügen von Ansprüchen](#).

So konfigurieren Sie ein EMR Studio als SAML-Anwendung in Ihrem IdP-Portal

Sie können ein bestimmtes EMR Studio mithilfe eines Deep-Links zum Studio als SAML-Anwendung konfigurieren. Auf diese Weise können sich Benutzer bei Ihrem IdP-Portal anmelden und ein bestimmtes Studio starten, anstatt durch die Amazon-EMR-Konsole zu navigieren.

- Verwenden Sie das folgende Format, um nach der SAML-Assertion-Überprüfung einen Deep-Link zu Ihrem EMR Studio als Landing-URL zu konfigurieren.

```
https://console.aws.amazon.com/emr/home?region=<aws-region>#studio/<your-studio-id>/start
```

Richten Sie den IAM-Identity-Center-Authentifizierungsmodus für Amazon EMR Studio ein

Um sich auf EMR Studio AWS IAM Identity Center vorzubereiten, müssen Sie Ihre Identitätsquelle konfigurieren und Benutzer und Gruppen bereitstellen. Bei der Bereitstellung werden Benutzer- und Gruppeninformationen für die Verwendung durch IAM Identity Center und durch Anwendungen, die IAM Identity Center verwenden, zur Verfügung gestellt. Weitere Informationen finden Sie unter [Benutzer- und Gruppenbereitstellung](#).

EMR Studio unterstützt die Verwendung der folgenden Identitätsanbieter für IAM Identity Center:

- AWS Managed Microsoft AD und selbstverwaltetes Active Directory — Weitere Informationen finden Sie unter [Connect Ihrem Microsoft AD-Verzeichnis](#) herstellen.
- SAML-basierte Anbieter – Eine vollständige Liste finden Sie unter [Unterstützte Identitätsanbieter](#).
- Das IAM-Identity-Center-Verzeichnis – Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

So richten Sie IAM Identity Center für EMR Studio ein

1. Um IAM Identity Center für EMR Studio einzurichten, benötigen Sie Folgendes:
 - Ein Verwaltungskonto in Ihrer AWS Organisation, wenn Sie mehrere Konten in Ihrer Organisation verwenden.

Note

Sie sollten Ihr Verwaltungskonto nur verwenden, um IAM Identity Center zu aktivieren und Benutzer und Gruppen bereitzustellen. Nachdem Sie IAM Identity Center eingerichtet haben, verwenden Sie ein Mitgliedskonto, um ein EMR Studio zu erstellen und Benutzer und Gruppen zuzuweisen. Weitere Informationen zur AWS Terminologie finden Sie unter [AWS Organizations Terminologie und Konzepte](#).

- Wenn Sie IAM Identity Center vor dem 25. November 2019 aktiviert haben, müssen Sie möglicherweise Anwendungen aktivieren, die IAM Identity Center für die Konten in Ihrer AWS Organisation verwenden. Weitere Informationen finden Sie unter [Aktivieren von IAM Identity Center-integrierten](#) Anwendungen in Konten. AWS
 - Stellen Sie sicher, dass die Voraussetzungen auf der Seite der Voraussetzungen für [IAM Identity Center](#) aufgeführt sind.
2. Folgen Sie den Anweisungen unter [IAM Identity Center](#) aktivieren, um IAM Identity Center dort zu aktivieren, AWS-Region wo Sie das EMR Studio erstellen möchten.
 3. Verbinden Sie IAM Identity Center mit Ihrem Identitätsanbieter und stellen Sie die Benutzer und Gruppen bereit, die Sie dem Studio zuweisen möchten.

Wenn Sie ...	Vorgehensweise
Ein Microsoft-AD-Verzeichnis verwenden	1. Folgen Sie den Anweisungen unter Connect zu Ihrem Microsoft AD-Verzeichnis herstellen, um Ihr selbstverwaltetes Active Directory oder AWS Managed Microsoft AD Verzeichnis mithilfe von AWS Directory Service zu verbinden. 2. Um Benutzer und Gruppen für IAM Identity Center bereitzustellen, können Sie Identitätsdaten aus Ihrem Quell-AD mit IAM Identity Center synchronisieren. Sie können Identitäten aus Ihrem Quell-AD auf viele Arten synchronisieren. Eine Möglichkeit besteht darin, AD-Benutzer oder -Gruppen einem AWS -Konto in Ihrer

Wenn Sie ...	Vorgehensweise
	Organisation zuzuweisen. Anweisungen finden Sie unter Single Sign-On. Die Synchronisation kann bis zu zwei Stunden dauern. Nach diesem Schritt werden synchronisierte Benutzer und Gruppen in Ihrem Identitätsspeicher angezeigt.  Note Benutzer und Gruppen werden erst in Ihrem Identity Store angezeigt, wenn Sie Benutzer- und Gruppeninformationen synchronisieren oder die Benutzerverwaltung just-in-time (JIT) verwenden. Weitere Informationen finden Sie unter Bereitstellung, wenn Benutzer von Active Directory kommen. 3. (Optional) Nachdem Sie AD-Benutzer und -Gruppen synchronisiert haben, können Sie ihnen den Zugriff auf Ihr AWS Konto, das Sie im vorherigen Schritt konfiguriert haben, entziehen. Anweisungen finden Sie unter Benutzerzugriff entfernen.
Ein externer Identitätsanbieter	Folgen Sie den Anweisungen unter Verbindung zu Ihrem externen Identitätsanbieter herstellen.

Wenn Sie ...	Vorgehensweise
IAM-Identity-Center-Verzeichnis	Wenn Sie Benutzer und Gruppen in IAM Identity Center erstellen, erfolgt die Bereitstellung automatisch. Weitere Informationen finden Sie unter Identitäten im IAM Identity Center verwalten .

Sie können jetzt Benutzer und Gruppen aus Ihrem Identity Store einem EMR Studio zuweisen. Detaillierte Anweisungen finden Sie unter [Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu](#).

Eine EMR-Studio-Servicerolle erstellen

Über die EMR-Studio-Servicerolle

Jedes EMR Studio verwendet eine IAM-Rolle mit Berechtigungen, die es dem Studio ermöglichen, mit anderen AWS Diensten zu interagieren. Diese Servicerolle muss Berechtigungen beinhalten, die es EMR Studio ermöglichen, einen sicheren Netzwerkkanal zwischen Workspaces und Clustern einzurichten, Notebook-Dateien darin zu speichern und darauf zuzugreifen Amazon S3 Control, AWS Secrets Manager während ein Workspace mit einem Git-Repository verknüpft wird.

Verwenden Sie die Studio-Servicerolle (anstelle von Sitzungsrichtlinien), um alle Amazon-S3-Zugriffsberechtigungen für das Speichern von Notebookdateien und AWS Secrets Manager Zugriffsberechtigungen zu definieren.

So erstellen Sie eine Servicerolle für EMR Studio auf Amazon EC2 oder Amazon EKS

1. Folgen Sie den Anweisungen unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS Dienst, um die Servicerolle](#) mit der folgenden Vertrauensrichtlinie zu erstellen.

Important

Die folgende Vertrauensrichtlinie umfasst die Schlüssel [aws:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel, um die Berechtigungen zu beschränken, die Sie EMR Studio auf bestimmte Ressourcen in Ihrem Konto gewähren. Auf diese Weise können Sie sich vor dem [Problem des verwirrten Stellvertreters](#) schützen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
        }
      }
    }
  ]
}
```

2. Entfernen Sie die standardmäßigen Rollenberechtigungen. Fügen Sie dann die Berechtigungen aus der folgenden Beispiel-IAM-Berechtigungsrichtlinie hinzu. Alternativ können Sie eine benutzerdefinierte Richtlinie erstellen, die das [Berechtigungen für EMR-Studio-Servicerollen](#) verwendet.

Important

- Damit die EC2 Tag-basierte Zugriffskontrolle von Amazon mit EMR Studio funktioniert, müssen Sie den Zugriff für die `ModifyNetworkInterfaceAttribute` API wie in der folgenden Richtlinie dargestellt einrichten.
- Damit EMR Studio mit der Servicerolle funktioniert, dürfen Sie die folgenden Anweisungen nicht ändern:
`AllowAddingEMRTagsDuringDefaultSecurityGroupCreation` und `AllowAddingTagsDuringEC2ENICreation`.
- Um die Beispielrichtlinie verwenden zu können, müssen Sie die folgenden Ressourcen mit dem Schlüssel **"for-use-with-amazon-emr-managed-policies"** und dem Wert **"true"** kennzeichnen.

- Ihre Amazon Virtual Private Cloud (VPC) für EMR Studio.
- Jedes Subnetz, das Sie mit dem Studio verwenden möchten.
- Alle benutzerdefinierten EMR-Studio-Sicherheitsgruppen. Sie müssen alle Sicherheitsgruppen, die Sie während der Vorschauphase von EMR Studio erstellt haben, taggen, wenn Sie sie weiterhin verwenden möchten.
- Geheimnisse AWS Secrets Manager, die in Studio-Benutzern verwaltet werden, um Git-Repositorys mit einem Workspace zu verknüpfen.

Sie können Tags auf Ressourcen anwenden, indem Sie die Registerkarte Tags auf dem entsprechenden Ressourcenbildschirm in verwenden. AWS Management Console

Falls zutreffend, ändern Sie `*` in der folgenden Richtlinie zu `"Resource": "*"` , um den Amazon-Ressourcennamen (ARN) der Ressource oder Ressourcen anzugeben, für die die Anweisung für Ihre Anwendungsfälle gilt bzw. gelten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEMRReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowEC2ENIActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ]
    }
  ]
}
```

```

    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
      }
    },
    {
      "Sid": "AllowEC2ENIAttributeAction",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifyNetworkInterfaceAttribute"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid": "AllowEC2SecurityGroupActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2>DeleteNetworkInterfacePermission"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },

```

```

    "Condition": {
      "StringEquals": {
        "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
          "ec2:CreateAction": "CreateSecurityGroup"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],

```

```

    "Condition": {
      "StringEquals": {
        "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
      }
    },
    {
      "Sid": "AllowEC2ENICreationInSubnetAndSecurityGroupWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingTagsDuringEC2ENICreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        }
      }
    },
    {
      "Sid": "AllowEC2ReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeTags",
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets",

```

```

        "ec2:DescribeVpcs"
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
    "Effect": "Allow",
    "Action": [
      "secretsmanager:GetSecretValue"
    ],
    "Resource": "arn:aws:secretsmanager:*:*:secret:*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
      }
    }
  },
  {
    "Sid": "AllowWorkspaceCollaboration",
    "Effect": "Allow",
    "Action": [
      "iam:GetUser",
      "iam:GetRole",
      "iam:ListUsers",
      "iam:ListRoles",
      "sso:GetManagedApplicationInstance",
      "sso-directory:SearchUsers"
    ],
    "Resource": "*"
  }
]
}

```

3. Erteilen Sie Ihrer Servicerolle Lese- und Schreibzugriff auf Ihren Amazon-S3-Standort für EMR Studio. Verwenden Sie die folgenden Mindestberechtigungen. Für weitere Informationen finden Sie unter [Amazon S3: Gewährt Lese- und Schreibzugriff auf Objekte in einem S3-Bucket programmgesteuert und in der Konsole](#).

```

"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"

```

Wenn Sie Ihren Amazon-S3-Bucket verschlüsseln, fügen Sie die folgenden Berechtigungen für AWS Key Management Service hinzu.

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

4. Wenn Sie den Zugriff auf Git-Secrets auf Benutzerebene kontrollieren möchten, fügen Sie tagbasierte Berechtigungen in der EMR-Studio-Benutzerrollenrichtlinie zu `secretsmanager:GetSecretValue` hinzu und entfernen Sie die Berechtigungen für die `secretsmanager:GetSecretValue`-Richtlinie aus der EMR-Studio-Servicerollenrichtlinie. Weitere Informationen zum Festlegen von differenzierten Benutzerberechtigungen finden Sie unter [Berechtigungsrichtlinien für EMR-Studio-Benutzer erstellen](#).

Minimale Servicerolle für EMR Serverless

Wenn Sie interaktive Workloads mit EMR Serverless über EMR-Studio-Notebooks ausführen möchten, verwenden Sie dieselbe Vertrauensrichtlinie, die Sie für die Einrichtung von EMR Studio im vorherigen Abschnitt [So erstellen Sie eine Servicerolle für EMR Studio auf Amazon EC2 oder Amazon EKS](#) verwendet haben.

Für Ihre IAM-Richtlinie verfügt die Mindestrichtlinie über die folgenden Berechtigungen. Aktualisieren Sie *bucket-name* mit dem Namen des Buckets, den Sie bei der Konfiguration von EMR Studio und Workspace verwenden möchten. EMR Studio verwendet das Bucket-Backup der Workspaces und Notebookdateien in Ihrem Studio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ObjectActions",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": ["arn:aws:s3:::bucket-name/*"]
    }
  ]
}
```



```

    },
    {
      "Sid": "BucketActions",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetEncryptionConfiguration"
      ],
      "Resource": ["arn:aws:s3:::bucket-name"]
    }
  ]
}

```

Wenn Sie beabsichtigen, einen verschlüsselten Amazon-S3-Bucket zu verwenden, fügen Sie Ihrer Richtlinie die folgenden Berechtigungen hinzu:

```

"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"

```

Berechtigungen für EMR-Studio-Servicerollen

In der folgenden Tabelle sind die Operationen aufgeführt, die EMR Studio mithilfe der Servicerolle ausführt, zusammen mit den IAM-Aktionen, die für jeden Vorgang erforderlich sind.

Operation	Aktionen
Richten Sie einen sicheren Netzwerkkanal zwischen einem Workspace und einem EMR-Cluster ein und führen Sie die erforderlichen Bereinigungsaktionen durch.	<pre> "ec2:CreateNetworkInterface", "ec2:CreateNetworkInterfacePermission", "ec2>DeleteNetworkInterface", "ec2>DeleteNetworkInterfacePermission", "ec2:DescribeNetworkInterfaces", "ec2:ModifyNetworkInterfaceAttribute", "ec2:AuthorizeSecurityGroupEgress", "ec2:AuthorizeSecurityGroupIngress", "ec2:CreateSecurityGroup", "ec2:DescribeSecurityGroups", "ec2:RevokeSecurityGroupEgress", "ec2:DescribeTags", </pre>

Operation	Aktionen
	<code>"ec2:DescribeInstances", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "elasticmapreduce:ListInstances", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListSteps"</code>
Verwenden Sie die in gespeicherten Git-Anmeldeinformationen AWS Secrets Manager, um Git-Repositorys mit einem Workspace zu verknüpfen.	<code>"secretsmanager:GetSecretValue"</code>
Wenden Sie AWS Tags auf die Netzwerkschnittstelle und die Standardsicherheitsgruppen an, die EMR Studio beim Einrichten des sicheren Netzwerkkanals erstellt. Weitere Informationen finden Sie unter Markieren von AWS -Ressourcen .	<code>"ec2:CreateTags"</code>

Operation	Aktionen
Greifen Sie auf Notebook-Dateien und Metadaten zu oder laden Sie sie in Amazon S3 hoch.	<pre data-bbox="690 241 1209 441">"s3:PutObject", "s3:GetObject", "s3:GetEncryptionConfiguration", "s3:ListBucket", "s3:DeleteObject"</pre> Wenn Sie einen verschlüsselten Amazon-S3-Bucket verwenden, schließen Sie die folgenden Berechtigungen ein. <pre data-bbox="690 682 1047 882">"kms:Decrypt", "kms:GenerateDataKey", "kms:ReEncryptFrom", "kms:ReEncryptTo", "kms:DescribeKey"</pre>
Aktivieren und konfigurieren Sie die Workspace-Zusammenarbeit.	<pre data-bbox="690 955 1274 1281">"iam:GetUser", "iam:GetRole", "iam:ListUsers", "iam:ListRoles", "sso:GetManagedApplicationInstance", "sso-directory:SearchUsers", "sso:DescribeApplication", "sso:DescribeInstance"</pre>
Verschlüsseln Sie EMR Studio Workspace-Notizbücher und -Dateien mithilfe von vom Kunden verwalteten Schlüsseln (CMK) mit AWS Key Management Service	<pre data-bbox="690 1354 1047 1554">"kms:Decrypt", "kms:GenerateDataKey", "kms:ReEncryptFrom", "kms:ReEncryptTo", "kms:DescribeKey"</pre>

EMR Studio-Benutzerberechtigungen für Amazon EC2 oder Amazon EKS konfigurieren

Sie müssen Benutzerberechtigungsrichtlinien für Amazon EMR Studio konfigurieren, damit Sie detaillierte Benutzer- und Gruppenberechtigungen festlegen können. Informationen zur Funktionsweise von Benutzerberechtigungen in EMR Studio finden Sie unter [Zugriffskontrolle](#) unter [Wie Amazon EMR Studio funktioniert](#).

Note

Die in diesem Abschnitt behandelten Berechtigungen erzwingen keine Datenzugriffskontrolle. Um den Zugriff auf Eingabe-Datensätze zu verwalten, sollten Sie Berechtigungen für die Cluster konfigurieren, die Ihr Studio verwendet. Weitere Informationen finden Sie unter [Sicherheit in Amazon EMR](#).

Eine EMR-Studio-Benutzerrolle für den IAM-Identity-Center-Authentifizierungsmodus erstellen

Sie müssen eine EMR-Studio-Benutzerrolle erstellen, wenn Sie den IAM-Identity-Center-Authentifizierungsmodus verwenden.

So erstellen Sie eine Benutzerrolle für EMR Studio

1. Folgen Sie den Anweisungen unter [Erstellen einer Rolle zum Delegieren von Berechtigungen für einen AWS Dienst](#) im Benutzerhandbuch, um eine AWS Identity and Access Management Benutzerrolle zu erstellen.

Verwenden Sie beim Erstellen der Rolle die folgende Vertrauensbeziehungsrichtlinie.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetContext"
      ]
    }
  ]
}
```

```
}  
]  
}
```

2. Entfernen Sie die standardmäßigen Rollenberechtigungen und -richtlinien.
3. Bevor Sie einem Studio Benutzer oder Gruppen zuweisen, hängen Sie Ihre EMR-Studio-Sitzungsrichtlinien an die Benutzerrolle an. Anweisungen zum Erstellen von Sitzungsrichtlinien finden Sie unter [Berechtigungsrichtlinien für EMR-Studio-Benutzer erstellen](#).

Berechtigungsrichtlinien für EMR-Studio-Benutzer erstellen

In den folgenden Abschnitten finden Sie Informationen zum Erstellen von Berechtigungsrichtlinien für EMR Studio.

Themen

- [Die Berechtigungsrichtlinien erstellen](#)
- [Legen Sie die Eigentümerschaft für die Workspace-Zusammenarbeit fest](#)
- [Git-Secrets-Richtlinie auf Benutzerebene erstellen](#)
- [Die Berechtigungsrichtlinie an Ihre IAM-Identität anhängen](#)

Note

Um Amazon-S3-Zugriffsberechtigungen für die Speicherung von Notebook-Dateien und AWS Secrets Manager -Zugriffsberechtigungen für das Lesen von Geheimnissen beim Verknüpfen von Workspaces mit Git-Repositories festzulegen, verwenden Sie die EMR-Studio-Servicerolle.

Die Berechtigungsrichtlinien erstellen

Erstellen Sie eine oder mehrere IAM-Berechtigungsrichtlinien, die festlegen, welche Aktionen ein Benutzer in Ihrem Studio ausführen kann. Mit den Beispielrichtlinien auf dieser Seite können Sie beispielsweise drei separate Richtlinien für [einfache](#), [fortgeschrittene](#) und [erfahrene](#) Studio-Benutzer erstellen.

Eine Aufschlüsselung aller Studio-Vorgänge, die ein Benutzer möglicherweise ausführen kann, und der IAM-Aktionen, die zur Ausführung dieses Vorgangs mindestens erforderlich sind, finden Sie

unter [AWS Identity and Access Management Berechtigungen für EMR Studio-Benutzer](#). Schritte zum Erstellen von Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Ihre Berechtigungsrichtlinie muss die folgenden Aussagen enthalten.

```
{
    "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
    "Effect": "Allow",
    "Action": "secretsmanager:TagResource",
    "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
},
{
    "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
    "Action": "iam:PassRole",
    "Resource": [
        "arn:aws:iam:*:*:role/your-emr-studio-service-role"
    ],
    "Effect": "Allow"
}
```

Legen Sie die Eigentümerschaft für die Workspace-Zusammenarbeit fest

Mithilfe von Workspace Collaboration können mehrere Benutzer gleichzeitig im selben Workspace arbeiten. Sie kann über das Collaboration-Bedienfeld in der Workspace-Benutzeroberfläche konfiguriert werden. Um das Collaboration Panel sehen und verwenden zu können, muss ein Benutzer über die folgenden Berechtigungen verfügen. Jeder Benutzer mit diesen Berechtigungen kann das Panel Zusammenarbeit sehen und verwenden.

```
"elasticmapreduce:UpdateEditor",
"elasticmapreduce:PutWorkspaceAccess",
"elasticmapreduce>DeleteWorkspaceAccess",
"elasticmapreduce:ListWorkspaceAccessIdentities"
```

Um den Zugriff auf das Panel Zusammenarbeit einzuschränken, können Sie die tagbasierte Zugriffskontrolle verwenden. Wenn ein Benutzer einen Workspace erstellt, wendet EMR Studio ein Standard-Tag mit einem Schlüssel von `creatorUserId`, dessen Wert die ID des Benutzers ist, der den Workspace erstellt.

Note

EMR Studio fügt das `creatorUserId`-Tag zu Workspaces hinzu, die nach dem dem 16. November 2021 erstellt wurden. Um einzuschränken, wer die Zusammenarbeit für vor dem Datum erstellte Workspaces konfigurieren kann, empfehlen wir, das `creatorUserId`-Tag manuell zu Ihrem Workspace hinzuzufügen und dann die tagbasierte Zugriffskontrolle in Ihren Benutzerberechtigungsrichtlinien zu verwenden.

Die folgende Beispielanweisung ermöglicht es einem Benutzer, die Zusammenarbeit für jeden Workspace mit dem Tag-Schlüssel `creatorUserId` zu konfigurieren, dessen Wert der Benutzer-ID entspricht (angegeben durch die RichtlinienvARIABLE `aws:userId`). Mit anderen Worten, die Anweisung ermöglicht es einem Benutzer, die Zusammenarbeit für die von ihm erstellten Workspaces zu konfigurieren. Weitere Informationen zu RichtlinienvARIABLEN finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

```
{
  "Sid": "UserRolePermissionsForCollaboration",
  "Action": [
    "elasticmapreduce:UpdateEditor",
    "elasticmapreduce:PutWorkspaceAccess",
    "elasticmapreduce>DeleteWorkspaceAccess",
    "elasticmapreduce:ListWorkspaceAccessIdentities"
  ],
  "Resource": "*",
  "Effect": "Allow",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
    }
  }
}
```

Git-Secrets-Richtlinie auf Benutzerebene erstellen

Themen

- [So verwenden Sie Berechtigungen auf Benutzerebene](#)
- [So gehen Sie von Berechtigungen auf Serviceebene zu Berechtigungen auf Benutzerebene über](#)
- [Berechtigungen auf Serviceebene](#)

So verwenden Sie Berechtigungen auf Benutzerebene

EMR Studio fügt das `for-use-with-amazon-emr-managed-user-policies`-Tag automatisch hinzu, wenn es Git-Secrets erstellt. Wenn Sie den Zugriff auf Git-Secrets auf Benutzerebene kontrollieren möchten, fügen Sie der EMR-Studio-Benutzerrollenrichtlinie tagbasierte Berechtigungen mit `secretsmanager:GetSecretValue` hinzu, wie im folgenden [So gehen Sie von Berechtigungen auf Serviceebene zu Berechtigungen auf Benutzerebene über](#)-Abschnitt beschrieben.

Wenn Sie für `secretsmanager:GetSecretValue` in der EMR-Studio-Servicerollenrichtlinie bereits über Berechtigungen verfügen, sollten Sie diese Berechtigungen entfernen.

So gehen Sie von Berechtigungen auf Serviceebene zu Berechtigungen auf Benutzerebene über

Note

Das `for-use-with-amazon-emr-managed-user-policies`-Tag stellt sicher, dass die Berechtigungen aus Schritt 1 unten dem Ersteller des Workspace Zugriff auf das Git-Secret gewähren. Wenn Sie Git-Repositorys jedoch vor dem 1. September 2023 verlinkt haben, wird den entsprechenden Git-Secrets der Zugriff verweigert, da das `for-use-with-amazon-emr-managed-user-policies`-Tag nicht auf sie angewendet wurde. Um Berechtigungen auf Benutzerebene anzuwenden, müssen Sie die alten Geheimnisse aus den entsprechenden Git-Repositorys neu erstellen JupyterLab und die entsprechenden Git-Repositorys erneut verknüpfen.

Weitere Informationen zu Richtlinienvariablen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

1. Fügen Sie der [EMR-Studio-Benutzerrollenrichtlinie](#) die folgenden Berechtigungen hinzu. Sie verwendet den `for-use-with-amazon-emr-managed-user-policies`-Schlüssel mit dem Wert `"${aws:userid}"`.

```
{
  "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": "arn:aws:secretsmanager:*:*:secret:",
  "Condition": {
    "StringEquals": {
```



```

        "secretsmanager:ResourceTag/for-use-with-amazon-emr-managed-user-
policies": "${aws:userid}"
    }
}

```

2. Falls vorhanden, entfernen Sie die folgende Berechtigung aus der [EMR-Studio-Service Rollenrichtlinie](#). Da die Service Rollenrichtlinie für alle von jedem Benutzer definierten Secrets gilt, müssen Sie dies nur einmal tun.

```

{
  "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
  "Effect": "Allow",
  "Action": [
    "secretsmanager:GetSecretValue"
  ],
  "Resource": "arn:aws:secretsmanager:*:*:secret:*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
}

```

Berechtigungen auf Serviceebene

Ab dem 1. September 2023 fügt EMR Studio das `for-use-with-amazon-emr-managed-user-policies`-Tag für die Zugriffskontrolle auf Benutzerebene automatisch hinzu. Da es sich um eine zusätzliche Funktion handelt, können Sie weiterhin den Zugriff auf Serviceebene verwenden, der über die `GetSecretValue`-Berechtigung in der [EMR-Studio-Service Rolle](#) verfügbar ist.

Für Secrets, die vor dem 1. September 2023 erstellt wurden, hat EMR Studio das `for-use-with-amazon-emr-managed-user-policies`-Tag nicht hinzugefügt. Um weiterhin Berechtigungen auf Serviceebene zu verwenden, behalten Sie einfach Ihre bestehenden [EMR-Studio-Service Rollen](#) und Benutzerrollenberechtigungen bei. Um jedoch einzuschränken, wer auf ein einzelnes Secret zugreifen kann, empfehlen wir, dass Sie die Schritte unter [So verwenden Sie Berechtigungen auf Benutzerebene](#) zum Hinzufügen des `for-use-with-amazon-emr-managed-user-policies`-Tags zu Ihren Secrets befolgen und dann die tagbasierte Zugriffskontrolle in Ihren Benutzerberechtigungsrichtlinien verwenden.

Weitere Informationen zu Richtlinienvariablen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

Die Berechtigungsrichtlinie an Ihre IAM-Identität anhängen

In der folgenden Tabelle wird zusammengefasst, an welche IAM-Identität Sie je nach Ihrem EMR-Studio-Authentifizierungsmodus eine Berechtigungsrichtlinie anhängen. Anweisungen zum Anfügen einer Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Wenn Sie ...	Fügen Sie die Richtlinie an ...
IAM-Authentifizierung	Ihre IAM-Identitäten (Benutzer, Benutzergruppen oder Rollen). Sie können einem Benutzer in Ihrem AWS-Konto eine Berechtigungsrichtlinie zuweisen.
IAM-Verbund mit einem externen Identitätsanbieter (IdP)	Die IAM-Rolle oder Rollen, die Sie für Ihren externen IdP erstellen. Zum Beispiel ein IAM für den SAML-2.0-Verbund. EMR Studio verwendet die Berechtigungen, die Sie Ihren IAM-Rollen zuordnen, für Benutzer mit Verbundzugriff auf ein Studio.
IAM Identity Center	Ihre Amazon-EMR-Studio-Benutzerrolle.

Beispielbenutzerrichtlinien

Die folgende grundlegende Benutzerrichtlinie erlaubt die meisten EMR-Studio-Aktionen, erlaubt es einem Benutzer jedoch nicht, neue Amazon-EMR-Cluster zu erstellen.

Grundlegende Richtlinien

Important

Die Beispielrichtlinie beinhaltet nicht die `CreateStudioPresignedUrl`-Berechtigung, die Sie einem Benutzer gewähren müssen, wenn Sie den IAM-Authentifizierungsmodus verwenden. Weitere Informationen finden Sie unter [Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu](#).

Die Beispielrichtlinie enthält Condition-Elemente zur Durchsetzung der tagbasierten Zugriffskontrolle (TBAC), sodass Sie die Richtlinie mit der Beispielservicerolle für EMR Studio verwenden können. Weitere Informationen finden Sie unter [Eine EMR-Studio-Servicerolle erstellen](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
          "ec2:CreateAction": "CreateSecurityGroup"
        }
      }
    },
    {
      "Sid": "AllowSecretManagerListSecrets",
      "Action": [
        "secretsmanager:ListSecrets"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

```

    },
    {
      "Sid": "AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
      "Effect": "Allow",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
      "Effect": "Allow",
      "Action": "secretsmanager:TagResource",
      "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
    },
    {
      "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam:*:*:role/<your-emr-studio-service-role>"
      ],
      "Effect": "Allow"
    },
    {
      "Sid": "AllowS3ListAndLocationPermissions",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": "arn:aws:s3:::*",
      "Effect": "Allow"
    },
    {
      "Sid": "AllowS3ReadOnlyAccessToLogs",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
      ]
    },
  ],

```

```

    "Effect": "Allow"
  },
  {
    "Sid": "AllowConfigurationForWorkspaceCollaboration",
    "Action": [
      "elasticmapreduce:UpdateEditor",
      "elasticmapreduce:PutWorkspaceAccess",
      "elasticmapreduce>DeleteWorkspaceAccess",
      "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
      "StringEquals": {
        "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
      }
    }
  },
  {
    "Sid": "DescribeNetwork",
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcs",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ListIAMRoles",
    "Effect": "Allow",
    "Action": [
      "iam:ListRoles"
    ],
    "Resource": "*"
  }
]
}

```

Die folgende Richtlinie für Zwischenbenutzer ermöglicht die meisten EMR-Studio-Aktionen und ermöglicht es einem Benutzer, mithilfe einer Clustervorlage neue Amazon-EMR-Cluster zu erstellen.

Zwischenrichtlinie

Important

Die Beispielrichtlinie beinhaltet nicht die `CreateStudioPresignedUrl`-Berechtigung, die Sie einem Benutzer gewähren müssen, wenn Sie den IAM-Authentifizierungsmodus verwenden. Weitere Informationen finden Sie unter [Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu](#).

Die Beispielrichtlinie enthält `Condition`-Elemente zur Durchsetzung der tagbasierten Zugriffskontrolle (TBAC), sodass Sie die Richtlinie mit der Beispielservicerolle für EMR Studio verwenden können. Weitere Informationen finden Sie unter [Eine EMR-Studio-Servicerolle erstellen](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEMRBasicActions",
      "Action": [
        "elasticmapreduce:CreateEditor",
        "elasticmapreduce:DescribeEditor",
        "elasticmapreduce:ListEditors",
        "elasticmapreduce:StartEditor",
        "elasticmapreduce:StopEditor",
        "elasticmapreduce>DeleteEditor",
        "elasticmapreduce:OpenEditorInConsole",
        "elasticmapreduce:AttachEditor",
        "elasticmapreduce:DetachEditor",
        "elasticmapreduce:CreateRepository",
        "elasticmapreduce:DescribeRepository",
        "elasticmapreduce>DeleteRepository",
        "elasticmapreduce:ListRepositories",
        "elasticmapreduce:LinkRepository",
        "elasticmapreduce:UnlinkRepository",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:ListClusters",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:CreatePersistentAppUI",
        "elasticmapreduce:DescribePersistentAppUI",

```

```

        "elasticmapreduce:GetPersistentAppUIPresignedURL",
        "elasticmapreduce:GetOnClusterAppUIPresignedURL"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Sid": "AllowEMRContainersBasicActions",
    "Action": [
        "emr-containers:DescribeVirtualCluster",
        "emr-containers:ListVirtualClusters",
        "emr-containers:DescribeManagedEndpoint",
        "emr-containers:ListManagedEndpoints",
        "emr-containers:DescribeJobRun",
        "emr-containers:ListJobRuns"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Sid": "AllowRetrievingManagedEndpointCredentials",
    "Effect": "Allow",
    "Action": [
        "emr-containers:GetManagedEndpointSessionCredentials"
    ],
    "Resource": [
        "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
    ],
    "Condition": {
        "StringEquals": {
            "emr-containers:ExecutionRoleArn": [
                "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
            ]
        }
    }
},
{
    "Sid": "AllowSecretManagerListSecrets",
    "Action": [
        "secretsmanager:ListSecrets"
    ],
    "Resource": "*",
    "Effect": "Allow"
}

```

```

    },
    {
      "Sid": "AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
      "Effect": "Allow",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
      "Effect": "Allow",
      "Action": "secretsmanager:TagResource",
      "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
    },
    {
      "Sid": "AllowClusterTemplateRelatedIntermediateActions",
      "Action": [
        "servicecatalog:DescribeProduct",
        "servicecatalog:DescribeProductView",
        "servicecatalog:DescribeProvisioningParameters",
        "servicecatalog:ProvisionProduct",
        "servicecatalog:SearchProducts",
        "servicecatalog:UpdateProvisionedProduct",
        "servicecatalog:ListProvisioningArtifacts",
        "servicecatalog:ListLaunchPaths",
        "servicecatalog:DescribeRecord",
        "cloudformation:DescribeStackResources"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam:*:*:role/<your-emr-studio-service-role>"
      ],
      "Effect": "Allow"
    },
    {

```



```

    "Sid": "AllowS3ListAndLocationPermissions",
    "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::*",
    "Effect": "Allow"
},
{
    "Sid": "AllowS3ReadOnlyAccessToLogs",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
    ],
    "Effect": "Allow"
},
{
    "Sid": "AllowConfigurationForWorkspaceCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
        "elasticmapreduce>DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
        }
    }
},
{
    "Sid": "DescribeNetwork",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
}

```

```

    },
    {
      "Sid": "ListIAMRoles",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowServerlessActions",
      "Action": [
        "emr-serverless:CreateApplication",
        "emr-serverless:UpdateApplication",
        "emr-serverless>DeleteApplication",
        "emr-serverless:ListApplications",
        "emr-serverless:GetApplication",
        "emr-serverless:StartApplication",
        "emr-serverless:StopApplication",
        "emr-serverless:StartJobRun",
        "emr-serverless:CancelJobRun",
        "emr-serverless:ListJobRuns",
        "emr-serverless:GetJobRun",
        "emr-serverless:GetDashboardForJobRun",
        "emr-serverless:AccessInteractiveEndpoints"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
      "Effect": "Allow"
    }
  ]
}

```

Die folgende erweiterte Benutzerrichtlinie erlaubt alle EMR-Studio-Aktionen und ermöglicht es einem Benutzer, mithilfe einer Cluster-Vorlage oder durch Bereitstellung einer Cluster-Konfiguration neue Amazon-EMR-Cluster zu erstellen.

Erweiterte Richtlinien

Important

Die Beispielrichtlinie beinhaltet nicht die `CreateStudioPresignedUrl`-Berechtigung, die Sie einem Benutzer gewähren müssen, wenn Sie den IAM-Authentifizierungsmodus verwenden. Weitere Informationen finden Sie unter [Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu](#).

Die Beispielrichtlinie enthält `Condition`-Elemente zur Durchsetzung der tagbasierten Zugriffskontrolle (TBAC), sodass Sie die Richtlinie mit der Beispielservicerolle für EMR Studio verwenden können. Weitere Informationen finden Sie unter [Eine EMR-Studio-Servicerolle erstellen](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEMRBasicActions",
      "Action": [
        "elasticmapreduce:CreateEditor",
        "elasticmapreduce:DescribeEditor",
        "elasticmapreduce:ListEditors",
        "elasticmapreduce:StartEditor",
        "elasticmapreduce:StopEditor",
        "elasticmapreduce>DeleteEditor",
        "elasticmapreduce:OpenEditorInConsole",
        "elasticmapreduce:AttachEditor",
        "elasticmapreduce:DetachEditor",
        "elasticmapreduce:CreateRepository",
        "elasticmapreduce:DescribeRepository",
        "elasticmapreduce>DeleteRepository",
        "elasticmapreduce:ListRepositories",
        "elasticmapreduce:LinkRepository",
        "elasticmapreduce:UnlinkRepository",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:ListClusters",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:CreatePersistentAppUI",
        "elasticmapreduce:DescribePersistentAppUI",

```

```

        "elasticmapreduce:GetPersistentAppUIPresignedURL",
        "elasticmapreduce:GetOnClusterAppUIPresignedURL"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Sid": "AllowEMRContainersBasicActions",
    "Action": [
        "emr-containers:DescribeVirtualCluster",
        "emr-containers:ListVirtualClusters",
        "emr-containers:DescribeManagedEndpoint",
        "emr-containers:ListManagedEndpoints",
        "emr-containers:DescribeJobRun",
        "emr-containers:ListJobRuns"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Sid": "AllowRetrievingManagedEndpointCredentials",
    "Effect": "Allow",
    "Action": [
        "emr-containers:GetManagedEndpointSessionCredentials"
    ],
    "Resource": [
        "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-  
cluster-id>/endpoints/<managed-endpoint-id>"
    ],
    "Condition": {
        "StringEquals": {
            "emr-containers:ExecutionRoleArn": [
                "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
            ]
        }
    }
},
{
    "Sid": "AllowSecretManagerListSecrets",
    "Action": [
        "secretsmanager:ListSecrets"
    ],
    "Resource": "*",
    "Effect": "Allow"
}

```

```

    },
    {
      "Sid": "AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
      "Effect": "Allow",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
      "Effect": "Allow",
      "Action": "secretsmanager:TagResource",
      "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
    },
    {
      "Sid": "AllowClusterTemplateRelatedIntermediateActions",
      "Action": [
        "servicecatalog:DescribeProduct",
        "servicecatalog:DescribeProductView",
        "servicecatalog:DescribeProvisioningParameters",
        "servicecatalog:ProvisionProduct",
        "servicecatalog:SearchProducts",
        "servicecatalog:UpdateProvisionedProduct",
        "servicecatalog:ListProvisioningArtifacts",
        "servicecatalog:ListLaunchPaths",
        "servicecatalog:DescribeRecord",
        "cloudformation:DescribeStackResources"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "AllowEMRCreatClusterAdvancedActions",
      "Action": [
        "elasticmapreduce:RunJobFlow"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {

```

```

    "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
    "Action": "iam:PassRole",
    "Resource": [
        "arn:aws:iam::*:role/<your-emr-studio-service-role>",
        "arn:aws:iam::*:role/EMR_DefaultRole_V2",
        "arn:aws:iam::*:role/EMR_EC2_DefaultRole"
    ],
    "Effect": "Allow"
},
{
    "Sid": "AllowS3ListAndLocationPermissions",
    "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation"
    ],
    "Resource": "arn:aws:s3:::*",
    "Effect": "Allow"
},
{
    "Sid": "AllowS3ReadOnlyAccessToLogs",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
    ],
    "Effect": "Allow"
},
{
    "Sid": "AllowConfigurationForWorkspaceCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
        "elasticmapreduce>DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
        }
    }
}

```

```

    },
    {
      "Sid" : "SageMakerDataWranglerForEMRStudio",
      "Effect" : "Allow",
      "Action" : [
        "sagemaker:CreatePresignedDomainUrl",
        "sagemaker:DescribeDomain",
        "sagemaker:ListDomains",
        "sagemaker:ListUserProfiles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeNetwork",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ListIAMRoles",
      "Effect": "Allow",
      "Action": [
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowServerlessActions",
      "Action": [
        "emr-serverless:CreateApplication",
        "emr-serverless:UpdateApplication",
        "emr-serverless>DeleteApplication",
        "emr-serverless:ListApplications",
        "emr-serverless:GetApplication",
        "emr-serverless:StartApplication",
        "emr-serverless:StopApplication",
        "emr-serverless:StartJobRun",
        "emr-serverless:CancelJobRun",
        "emr-serverless:ListJobRuns",
        "emr-serverless:GetJobRun",

```

```

        "emr-serverless:GetDashboardForJobRun",
        "emr-serverless:AccessInteractiveEndpoints"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
    "Effect": "Allow"
},
{
    "Sid": "AllowCodeWhisperer",
    "Effect": "Allow",
    "Action": [ "codewhisperer:GenerateRecommendations" ],
    "Resource": "*"
},
{
    "Sid": "AllowAthenaSQL",
    "Action": [
        "athena:StartQueryExecution",
        "athena:StopQueryExecution",
        "athena:GetQueryExecution",
        "athena:GetQueryRuntimeStatistics",
        "athena:GetQueryResults",
        "athena:ListQueryExecutions",
        "athena:BatchGetQueryExecution",
        "athena:GetNamedQuery",
        "athena:ListNamedQueries",
        "athena:BatchGetNamedQuery",
        "athena:UpdateNamedQuery",
        "athena>DeleteNamedQuery",
        "athena:ListDataCatalogs",
        "athena:GetDataCatalog",
        "athena:ListDatabases",
        "athena:GetDatabase",
        "athena:ListTableMetadata",
        "athena:GetTableMetadata",
        "athena:ListWorkGroups",
        "athena:GetWorkGroup",
        "athena:CreateNamedQuery",
        "athena:GetPreparedStatement",
        "glue:CreateDatabase",
    ]
}

```



```

        "glue:DeleteDatabase",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:UpdateDatabase",
        "glue:CreateTable",
        "glue:DeleteTable",
        "glue:BatchDeleteTable",
        "glue:UpdateTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:BatchCreatePartition",
        "glue:CreatePartition",
        "glue:DeletePartition",
        "glue:BatchDeletePartition",
        "glue:UpdatePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition",
        "kms:ListAliases",
        "kms:ListKeys",
        "kms:DescribeKey",
        "lakeformation:GetDataAccess",
        "s3:GetBucketLocation",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:AbortMultipartUpload",
        "s3:PutObject",
        "s3:PutBucketPublicAccessBlock",
        "s3:ListAllMyBuckets"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
}

```

Die folgende Benutzerrichtlinie enthält die Mindestbenutzerberechtigungen, die für die Verwendung einer interaktiven EMR-Serverless-Anwendung mit EMR Studio Workspaces erforderlich sind.

Interaktive EMR-Serverless-Richtlinie

Ersetzen Sie in dieser Beispielrychtlinie, die Benutzerberechtigungen für interaktive EMR Serverless-Anwendungen mit EMR Studio enthält, die Platzhalter für *serverless-runtime-role* und durch Ihre richtige EMR Studio-Dienstrolle und *emr-studio-service-role* EMR Serverless-Laufzeitrolle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowServerlessActions",
      "Action": [
        "emr-serverless:CreateApplication",
        "emr-serverless:UpdateApplication",
        "emr-serverless>DeleteApplication",
        "emr-serverless:ListApplications",
        "emr-serverless:GetApplication",
        "emr-serverless:StartApplication",
        "emr-serverless:StopApplication",
        "emr-serverless:StartJobRun",
        "emr-serverless:CancelJobRun",
        "emr-serverless:ListJobRuns",
        "emr-serverless:GetJobRun",
        "emr-serverless:GetDashboardForJobRun",
        "emr-serverless:AccessInteractiveEndpoints"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Sid": "AllowEMRBasicActions",
      "Action": [
        "elasticmapreduce:CreateEditor",
        "elasticmapreduce:DescribeEditor",
        "elasticmapreduce:ListEditors",
        "elasticmapreduce:UpdateStudio",
        "elasticmapreduce:StartEditor",
        "elasticmapreduce:StopEditor",
        "elasticmapreduce>DeleteEditor",
        "elasticmapreduce:OpenEditorInConsole",
        "elasticmapreduce:AttachEditor",
        "elasticmapreduce:DetachEditor",

```

```

        "elasticmapreduce:CreateStudio",
        "elasticmapreduce:DescribeStudio",
        "elasticmapreduce>DeleteStudio",
        "elasticmapreduce:ListStudios",
        "elasticmapreduce:CreateStudioPresignedUrl"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Sid": "AllowPassingRuntimeRoleForRunningEMRServerlessJob",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
    "Effect": "Allow"
},
{
    "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::*:role/emr-studio-service-role",
    "Effect": "Allow"
},
{
    "Sid": "AllowS3ListAndGetPermissions",
    "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:GetObject"
    ],
    "Resource": "arn:aws:s3:::*",
    "Effect": "Allow"
},
{
    "Sid": "DescribeNetwork",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
},
{
    "Sid": "ListIAMRoles",

```

```
        "Effect": "Allow",
        "Action": [
            "iam:ListRoles"
        ],
        "Resource": "*"
    }
]
```

AWS Identity and Access Management Berechtigungen für EMR Studio-Benutzer

Die folgende Tabelle enthält jeden Amazon-EMR-Studio-Vorgang, den ein Benutzer ausführen könnte, und listet die mindestens erforderlichen IAM-Aktionen auf, um diesen Vorgang auszuführen. Sie erlauben diese Aktionen in Ihren IAM-Berechtigungsrichtlinien (wenn Sie die IAM-Authentifizierung verwenden) oder in Ihren Benutzerrollen-Sitzungsrichtlinien (wenn Sie die IAM-Identity-Center-Authentifizierung verwenden) für EMR Studio.

In der Tabelle werden auch die Operationen angezeigt, die in den einzelnen Beispielberechtigungsrichtlinien für EMR Studio zulässig sind. Weitere Informationen zu Beispiel-Berechtigungsrichtlinien finden Sie unter [Berechtigungsrichtlinien für EMR-Studio-Benutzer erstellen](#).

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Arbeitsbereiche erstellen und löschen	Ja	Ja	Ja	"elasticmapreduce:CreateEditor", "elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:DeleteEditor"
Rufen Sie das Panel Zusammenarbeit auf, aktivieren Sie die Workspace-Zusammenarbeit und fügen Sie Mitarbeiter hinzu. Weitere Informationen finden	Ja	Ja	Ja	"elasticmapreduce:UpdateEditor", "elasticmapreduce:PutWorkspaceAccess", "elasticmapreduce:DeleteWorkspaceAccess",

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Sie unter Legen Sie die Eigentümerschaft für die Workspace-Zusammenarbeit fest.				"elasticmapreduce:ListWorkspaceAccessIdentities"
Sehen Sie sich beim Erstellen eines neuen EMR-Clusters eine Liste der Amazon S3 Control Speicher-Buckets in demselben Konto wie Studio an und greifen Sie auf Container-Logs zu, wenn Sie eine Web-UI zum Debuggen von Anwendungen verwenden	Ja	Ja	Ja	"s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetBucketLocation", "s3:GetObject"
Auf Workspaces zugreifen	Ja	Ja	Ja	"elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:StartEditor", "elasticmapreduce:StopEditor", "elasticmapreduce:OpenEditorInConsole"

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Vorhandene Amazon-EMR-Cluster, die mit dem Workspace verknüpft sind, anhängen oder trennen	Ja	Ja	Ja	<pre>"elasticmapreduce: AttachEditor", "elasticmapreduce:DetachEditor", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListInstanceGroups", "elasticmapreduce:ListBootstrapActions"</pre>
Amazon EMR in EKS-Clustern anfügen oder trennen	Ja	Ja	Ja	<pre>"elasticmapreduce: AttachEditor", "elasticmapreduce:DetachEditor", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListManagedEndpoints", "emr-containers:DescribeManagedEndpoint", "emr-containers:GetManagedEndpointSessionCredentials"</pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Serverless-EMR-Anwendungen, die dem Workspace zugeordnet sind, anhängen oder trennen	Nein	Ja	Ja	<pre>"elasticmapreduce: AttachEditor", "elasticmapreduce:DetachEditor", "emr-serverless:GetApplication", "emr-serverless:StartApplication", "emr-serverless:ListApplications", "emr-serverless:GetDashboardForJobRun", "emr-serverless:AccessInteractiveEndpoints", "iam:PassRole"</pre> Die PassRole-Berechtigung ist erforderlich, um die EMR-Serverless-Auftragsausführungs-Rolle zu übergeben. Weitere Informationen finden Sie unter Auftragslaufzeitrollen im Benutzerhandbuch für Amazon EMR Serverless.

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Debuggen Sie Amazon EMR bei EC2 Jobs mit persistenten Anwendungsbienutzeroberflächen	Ja	Ja	Ja	<pre>"elasticmapreduce: CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "elasticmapreduce:ListClusters", "elasticmapreduce:ListSteps", "elasticmapreduce:DescribeCluster", "s3:ListBucket", "s3:GetObject"</pre>
Debuggen Sie Amazon EMR bei EC2 Jobs mit Benutzeroberflächen für Cluster-Anwendungen	Ja	Ja	Ja	<pre>"elasticmapreduce: GetOnClusterAppUIPresignedURL"</pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Amazon EMR in EKS-Auftragsausführungen mit dem Spark History Server debuggen	Ja	Ja	Ja	<pre>"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListJobRuns", "emr-containers:DescribeJobRun", "s3:ListBucket", "s3:GetObject"</pre>
Git-Repositorys erstellen und löschen	Ja	Ja	Ja	<pre>"elasticmapreduce:CreateRepository", "elasticmapreduce>DeleteRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository", "secretsmanager:CreateSecret", "secretsmanager:ListSecrets", "secretsmanager:TagResource"</pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Git-Repositorys verknüpfen und trennen	Ja	Ja	Ja	<pre>"elasticmapreduce: LinkRepository", "elasticmapreduce:U nlinkRepository", "elasticmapreduce: ListRepositories", "elasticmapreduce:Describe Repository"</pre>
Neue Cluster aus vordefinierten Cluster-Vorlagen erstellen	Nein	Ja	Ja	<pre>"servicecatalog:Se archProducts", "servicecatalog:DescribePr oduct", "servicecatalog:Des cribeProductView", "servicecatalog:DescribePr ovisioningParameters", "servicecatalog:Provis ionProduct", "servicecatalog:UpdateP rovisionedProduct", "servicecatalog:ListProvi sioningArtifacts", "servicecatalog:DescribeRe cord", "servicecatalog:List LaunchPaths", "cloudformation:Descri beStackResources", "elasticmapreduce:ListClus ters", "elasticmapreduce:De scribeCluster"</pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Stellen Sie eine Clusterkonfiguration bereit, um neue Cluster zu erstellen.	Nein	Nein	Ja	<pre>"elasticmapreduce: RunJobFlow", "iam:PassRole", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"</pre>
Weisen Sie einem Studio einen Benutzer zu, wenn Sie den IAM-Authentifizierungsmodus verwenden.	Nein	Nein	Nein	<pre>"elasticmapreduce: CreateStudioPresignedUrl"</pre>
Beschreiben Sie Netzwerkobjekte.	Ja	Ja	Ja	<pre>{ "Version": "2012-10-17", "Statement": [{ "Sid": "DescribeNetwork", "Effect": "Allow", "Action": ["ec2:DescribeVpcs", "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups"], "Resource": "*" }] }</pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Listen Sie die IAM-Rollen auf.	Ja	Ja	Ja	<pre>{ "Version": "2012-10-17", "Statement": [{ "Sid": "ListIAMRoles", "Effect": "Allow", "Action": ["iam:ListRoles"], "Resource": "*" }] }</pre>
Stellen Sie von Amazon SageMaker AI Studio aus eine Connect zu EMR Studio her und verwenden Sie die visuelle Oberfläche von Data Wrangler.	Nein	Nein	Ja	<pre>"sagemaker:CreatePresignedDomainUrl", "sagemaker:DescribeDomain", "sagemaker:ListDomains", "sagemaker:ListUserProfileNames"</pre>
Verwenden Sie Amazon CodeWhisperer in Ihrem EMR Studio.	Nein	Nein	Ja	<pre>"codewhisperer:GenerateRecommendations"</pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
Greifen Sie von Ihrem EMR Studio aus auf den Amazon-Athena-SQL-Editor zu. Diese Liste enthält möglicherweise nicht alle Berechtigungen, die Sie für die Nutzung aller Athena-Features benötigen. Die up-to-date Liste der meisten finden Sie in der Athena-Vollzugsrichtlinie .	Nein	Nein	Ja	<pre> "athena:StartQueryExecution", "athena:StopQueryExecution", "athena:GetQueryExecution", "athena:GetQueryRuntimeStatistics", "athena:GetQueryResults", "athena:ListQueryExecutions", "athena:BatchGetQueryExecution", "athena:GetNamedQuery", "athena:ListNamedQueries", , "athena:BatchGetNamedQuery", "athena:UpdateNamedQuery", "athena>DeleteNamedQuery", "athena:ListDataCatalogs", "athena:GetDataCatalog", "athena:ListDatabases", "athena:GetDatabase", "athena:ListTableMetadata", "athena:GetTableMetadata", "athena:ListWorkGroups", "athena:GetWorkGroup", "athena:CreateNamedQuery", "athena:GetPreparedStatement", "glue:CreateDatabase", "glue>DeleteDatabase", "glue:GetDatabase", </pre>

Aktion	Basic	Intermediär	Advanced	Zugeordnete Aktionen
				"glue:GetDatabases", "glue:UpdateDatabase", "glue:CreateTable", "glue>DeleteTable", "glue:BatchDeleteTable", "glue:UpdateTable", "glue:GetTable", "glue:GetTables", "glue:BatchCreatePartition", "glue:CreatePartition", "glue>DeletePartition", "glue:BatchDeletePartitions", "glue:UpdatePartition", "glue:GetPartition", "glue:GetPartitions", "glue:BatchGetPartition", "kms:ListAliases", "kms:ListKeys", "kms:DescribeKey", "lakeformation:GetDataAccess", "s3:GetBucketLocation", "s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:ListBucketMultipartUploads", "s3:ListMultipartUploadParts", "s3:AbortMultipartUpload", "s3:PutObject", "s3:PutBucketPublicAccessBlock", "s3:ListAllMyBuckets"

Ein EMR Studio erstellen

Sie können ein EMR Studio für Ihr Team mithilfe der Amazon-EMR-Konsole oder der AWS CLI erstellen. Das Erstellen einer Studio-Instance ist Teil der Einrichtung von Amazon EMR Studio.

Voraussetzungen

Bevor Sie ein Studio erstellen, stellen Sie sicher, dass Sie die vorherigen Aufgaben in [Richten Sie ein EMR Studio ein](#) abgeschlossen haben.

Um ein Studio mit dem zu erstellen AWS CLI, sollten Sie die neueste Version installiert haben. Weitere Informationen finden Sie unter [Installieren oder Aktualisierung auf die neueste Version von AWS CLI](#).

Important

Deaktivieren Sie Proxy-Management-Tools wie FoxyProxy oder SwitchyOmega im Browser, bevor Sie ein Studio erstellen. Aktive Proxys können zu einer Netzwerkfehler-Fehlermeldung führen, wenn Sie Studio erstellen wählen.

Amazon EMR bietet Ihnen eine einfache Konsolenoberfläche zum Erstellen eines Studios, sodass Sie schnell mit den Standardeinstellungen beginnen können, um interaktive Workloads oder Batch-Jobs mit den Standardeinstellungen auszuführen. Wenn Sie ein EMR Studio erstellen, wird auch eine serverlose EMR-Anwendung erstellt, die für Ihre interaktiven Jobs bereit ist.

Wenn Sie die volle Kontrolle über die Einstellungen Ihres Studios haben möchten, können Sie Benutzerdefiniert wählen, wodurch Sie alle zusätzlichen Einstellungen konfigurieren können.

Interactive workloads

So erstellen Sie ein EMR Studio für interaktive Workloads

1. Öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie im linken Navigationsbereich unter EMR Studio die Option Erste Schritte aus. Sie können auf der Studio-Seite auch ein neues Studio erstellen.
3. Amazon EMR bietet Standardeinstellungen für Sie, wenn Sie ein EMR Studio für interaktive Workloads erstellen, aber Sie können diese Einstellungen bearbeiten. Zu den konfigurierbaren Einstellungen gehören der Name des EMR Studios, der S3-Speicherort für

Ihren Workspace, die zu verwendende Servicerolle, die Workspace (s), die Sie verwenden möchten, der Name der EMR Serverless-Anwendung und die zugehörige Runtime-Rolle.

4. Wählen Sie Create Studio und starten Sie Workspace, um den Vorgang abzuschließen und zur Studios-Seite zu navigieren. Ihr neues Studio wird in der Liste mit Details wie dem Studio-Namen, dem Erstellungsdatum und der Studio-Zugriffs-URL angezeigt. Ihr Workspace wird in einem neuen Tab in Ihrem Browser geöffnet.

Batch jobs

So erstellen Sie ein EMR Studio für interaktive Workloads

1. Öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie im linken Navigationsbereich unter EMR Studio die Option Erste Schritte aus. Sie können auf der Studio-Seite auch ein neues Studio erstellen.
3. Amazon EMR bietet Standardeinstellungen für Sie, wenn Sie ein EMR Studio für Batch-Jobs erstellen, aber Sie können diese Einstellungen bearbeiten. Zu den konfigurierbaren Einstellungen gehören der Name von EMR Studio, der Name der EMR Serverless-Anwendung und die zugehörige Runtime-Rolle.
4. Wählen Sie Create Studio und starten Sie Workspace, um den Vorgang abzuschließen und zur Studios-Seite zu navigieren. Ihr neues Studio wird in der Liste mit Details wie dem Studio-Namen, dem Erstellungsdatum und der Studio-Zugriffs-URL angezeigt. Ihr EMR Studio wird in einem neuen Tab in Ihrem Browser geöffnet.

Custom settings

Um ein EMR Studio mit benutzerdefinierten Einstellungen zu erstellen

1. Öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie im linken Navigationsbereich unter EMR Studio die Option Erste Schritte aus. Sie können auf der Studio-Seite auch ein neues Studio erstellen.
3. Wählen Sie Studio erstellen, um die Seite Studio erstellen zu öffnen.
4. Geben Sie einen Studio-Namen ein.
5. Wählen Sie, ob Sie einen neuen S3-Bucket erstellen oder einen vorhandenen Speicherort verwenden möchten.

6. Wählen Sie den Workspace aus, der dem Studio hinzugefügt werden soll. Sie können bis zu 3 Arbeitsbereiche hinzufügen.
7. Wählen Sie unter Authentifizierung einen Authentifizierungsmodus für das Studio und geben Sie die Informationen gemäß der folgenden Tabelle ein. Weitere Informationen zur Authentifizierung für EMR Studio finden Sie unter [Einen Authentifizierungsmodus für Amazon EMR Studio auswählen](#).

Wenn Sie ...	Vorgehensweise
IAM-Authentifizierung oder -Verbund verwenden	Die Standardauthentifizierungsmethode ist AWS Identity and Access Management (IAM). Am unteren Bildschirmrand können Sie auch Tags hinzufügen, um bestimmten Benutzern Zugriff auf das Studio zu gewähren, wie unter Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu beschrieben. Wenn Sie möchten, dass sich Verbundbenutzer mit der Studio-URL und den Anmeldeinformationen für Ihren Identitätsanbieter (IdP) anmelden, wählen Sie Ihren IdP aus der Dropdownliste aus und geben Sie Ihre Anmelde-URL und den Parameternamen Ihres Identitätsanbieters (IdP) ein. RelayState Eine Liste der IdP-Authentifizierung URLs und der RelayState Namen finden Sie unter RelayState Parameter und Authentifizierung des Identitätsanbieters URLs.
Authentifizierung von IAM Identity Center	Wählen Sie Ihre EMR-Studio-Servicerolle und Benutzerrolle aus. Weitere Informationen erhalten Sie unter Eine EMR-Studio-Servicerolle erstellen und Eine EMR-Studio-

Wenn Sie ...	Vorgehensweise
	o-Benutzerrolle für den IAM-Identity-Center-Authentifizierungsmodus erstellen. Wenn Sie die IAM Identity Center-Authentifizierung (früher AWS Single Sign On) für das Studio verwenden, können Sie die Anmeldung für Benutzer mit der Option Weitergabe vertrauenswürdiger Identitäten aktivieren optimieren. Mit Trusted Identity Propagation können sich Benutzer mit ihren Identity Center-Anmeldeinformationen anmelden und ihre Identitäten an nachgelagerte AWS Dienste weitergeben lassen, wenn sie das Studio verwenden. Im Abschnitt Application access (Anwendungszugriff) können Sie auch angeben, ob alle Benutzer und Gruppen in Ihrem Identity Center Zugriff auf das Studio haben sollen oder ob nur zugewiesene Benutzer und Gruppen, die Sie auswählen, auf das Studio zugreifen können. Weitere Informationen finden Sie unter Integrieren Sie Amazon EMR mit AWS IAM Identity Center und auch Trusted Identity Propagation zwischen Anwendungen im AWS IAM Identity Center-Benutzerhandbuch.

8. Wählen Sie für VPC eine Amazon Virtual Private Cloud (VPC) für das Studio aus der Dropdown-Liste aus.
9. Wählen Sie unter Subnetze maximal fünf Subnetze in Ihrer VPC aus, die Sie dem Studio zuordnen möchten. Sie haben die Möglichkeit, weitere Subnetze hinzuzufügen, nachdem Sie das Studio erstellt haben.

10. Wählen Sie für Sicherheitsgruppen entweder die Standardsicherheitsgruppen oder benutzerdefinierte Sicherheitsgruppen aus. Weitere Informationen finden Sie unter [Definieren Sie Sicherheitsgruppen zur Steuerung des Netzwerkverkehrs in EMR Studio](#).

Wenn Sie folgendes auswählen ...	Vorgehensweise
Die Standard-Sicherheitsgruppen von EMR Studio	Um die Git-basierte Repository-Verknüpfung für das Studio zu aktivieren, wählen Sie Cluster/Endpunkte und Git-Repository aktivieren. Wählen Sie andernfalls Cluster/Endpunkte aktivieren.
Benutzerdefinierte Sicherheitsgruppen für Ihr Studio	- Wählen Sie unter Cluster-/Endpunktsicherheitsgruppe die Engine-Sicherheitsgruppe aus, die Sie aus der Dropdownliste konfiguriert haben. Ihr Studio verwendet diese Sicherheitsgruppe, um eingehenden Zugriff von verbundenen Workspaces aus zu ermöglichen. - Wählen Sie unter Workspace-Sicherheitsgruppe die Workspace-Sicherheitsgruppe aus, die Sie aus der Dropdownliste konfiguriert haben. Ihr Studio verwendet diese Sicherheitsgruppe mit Workspaces, um ausgehenden Zugriff auf verbundene Amazon-EMR-Cluster und öffentlich gehostete Git-Repositorys zu ermöglichen.

11. Fügen Sie Ihrem Studio und anderen Ressourcen Tags hinzu. Weitere Informationen zu Tags finden Sie unter [Tag-Cluster](#).
12. Wählen Sie Create Studio und starten Sie Workspace, um den Vorgang abzuschließen und zur Studios-Seite zu navigieren. Ihr neues Studio wird in der Liste mit Details wie dem Studio-Namen, dem Erstellungsdatum und der Studio-Zugriffs-URL angezeigt.

Nachdem Sie ein Studio erstellt haben, folgen Sie den Anweisungen unter [Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu](#).

CLI

Note

Linux-Zeilenumbruchzeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

Example – Erstellen Sie ein EMR Studio, das IAM für die Authentifizierung verwendet

Der folgende AWS CLI Beispielfehl erstellt ein EMR Studio mit IAM-Authentifizierungsmodus. Wenn Sie die IAM-Authentifizierung oder den Verbund für das Studio verwenden, geben Sie kein `--user-role` an.

Damit sich Verbundbenutzer mit der Studio-URL und den Anmeldeinformationen für Ihren Identitätsanbieter (IdP) anmelden können, geben Sie Ihr `--idp-auth-url` und `--idp-relay-state-parameter-name` an. Eine Liste der IdP-Authentifizierung URLs und der RelayState Namen finden Sie unter [RelayState Parameter und Authentifizierung des Identitätsanbieters URLs](#).

```
aws emr create-studio \  
--name <example-studio-name> \  
--auth-mode IAM \  
--vpc-id <example-vpc-id> \  
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5> \  
--service-role <example-studio-service-role-name> \  
--user-role studio-user-role-name \  
--workspace-security-group-id <example-workspace-sg-id> \  
--engine-security-group-id <example-engine-sg-id> \  
--default-s3-location <example-s3-location> \  
--idp-auth-url <https://EXAMPLE/login/> \  
--idp-relay-state-parameter-name <example-RelayState>
```

Example – Erstellen Sie ein EMR Studio, das Identity Center für die Authentifizierung verwendet

Mit dem folgenden AWS CLI Beispielbefehl wird ein EMR Studio erstellt, das den IAM Identity Center-Authentifizierungsmodus verwendet. Wenn Sie die IAM-Identity-Center-Authentifizierung verwenden, müssen Sie eine `--user-role` angeben.

Weitere Informationen zum Authentifizierungsmodus von IAM Identity Center finden Sie unter [Richten Sie den IAM-Identity-Center-Authentifizierungsmodus für Amazon EMR Studio ein](#).

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode SSO \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5> \
--service-role <example-studio-service-role-name> \
--user-role <example-studio-user-role-name> \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location>
--trusted-identity-propagation-enabled \
--idc-user-assignment OPTIONAL \
--idc-instance-arn <iam-identity-center-instance-arn>
```

Example – CLI-Ausgabe für **aws emr create-studio**

Es folgt ein Beispiel für die Ausgabe, die nach dem Erstellen eines Studios erscheint.

```
{
  StudioId: "es-123XXXXXXXXX",
  Url: "https://es-123XXXXXXXXX.emrstudio-prod.us-east-1.amazonaws.com"
}
```

Weitere Informationen über den Befehl `create-studio` finden Sie unter [AWS CLI - Befehlsreferenz](#).

RelayState Parameter und Authentifizierung des Identitätsanbieters URLs

Wenn Sie den IAM-Verbund verwenden und möchten, dass sich Benutzer mit Ihrer Studio-URL und den Anmeldeinformationen für Ihren Identity Provider (IdP) anmelden, können Sie bei der Anmeldung Ihren Identity Provider (IdP) und den RelayStateParameternamen angeben. [Ein EMR Studio erstellen](#)

Die folgende Tabelle zeigt die Standard-Authentifizierungs-URL und den RelayState Parameternamen für einige beliebte Identitätsanbieter.

Identitätsanbieter	Parameter	Authentifizierungs-URL
Auth0	RelayState	<code>https://<sub_domain>.auth0.com/samlp/<app_id></code>
Google-Konten	RelayState	<code>https://accounts.google.com/o/saml2/initssso?idpid=<idp_id>&spid=<sp_id>&forceauthn=false</code>
Microsoft Azure	RelayState	<code>https://myapps.microsoft.com/signin/<app_name>/<app_id>?tenantId=<tenant_id></code>
Okta	RelayState	<code>https://<sub_domain>.okta.com/app/<app_name>/<app_id>/sso/saml</code>
PingFederate	TargetResource	<code>https://<host>/idp/<idp_id>/startSSO.ping?PartnerSpId=<sp_id></code>
PingOne	TargetResource	<code>https://sso.connect.pingidentity.com/sso/sp/initssso?saasid=<app_id>&idpid=<idp_id></code>

EMR-Studio-Benutzer zuweisen und verwalten

Nachdem Sie ein EMR Studio erstellt haben, können Sie ihm Benutzer und Gruppen zuweisen. Die Methode, mit der Sie Benutzer zuweisen, aktualisieren und entfernen, hängt vom Studio-Authentifizierungsmodus ab.

- Wenn Sie den IAM-Authentifizierungsmodus verwenden, konfigurieren Sie die Benutzerzuweisung und die Berechtigungen von EMR Studio in IAM oder mit IAM und Ihrem Identitätsanbieter.
- Im IAM Identity Center-Authentifizierungsmodus verwenden Sie die Amazon EMR-Verwaltungskonsolle oder die, AWS CLI um Benutzer zu verwalten.

Weitere Informationen zur Authentifizierung für Amazon EMR Studio finden Sie unter [Einen Authentifizierungsmodus für Amazon EMR Studio auswählen](#).

Weisen Sie EMR Studio einen Benutzer oder eine Gruppe zu

IAM

Wenn Sie [IAM-Authentifizierungsmodus für Amazon EMR Studio einrichten](#) verwenden, müssen Sie die `CreateStudioPresignedUrl`-Aktion in der IAM-Berechtigungsrichtlinie eines Benutzers zulassen und den Benutzer auf ein bestimmtes Studio beschränken. Sie können `CreateStudioPresignedUrl` in Ihre eigene [Benutzerberechtigungen für den IAM-Authentifizierungsmodus](#) aufnehmen oder eine separate Richtlinie verwenden.

Um einen Benutzer auf ein Studio (oder eine Gruppe von Studios) zu beschränken, können Sie die attributbasierte Zugriffskontrolle (ABAC) verwenden oder den Amazon-Ressourcennamen (ARN) eines Studios im Resource Element der Berechtigungsrichtlinie angeben.

Example Weisen Sie einem Studio mithilfe eines Studio-ARN einen Benutzer zu

Die folgende Beispielrichtlinie gewährt einem Benutzer Zugriff auf ein bestimmtes EMR Studio, indem die `CreateStudioPresignedUrl`-Aktion zugelassen und der Amazon-Ressourcenname (ARN) des Studios im Resource-Element angegeben wird.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateStudioPresignedUrl",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:CreateStudioPresignedUrl"
      ],
      "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/<studio-id>"
    }
  ]
}
```

Example Einem Studio mit ABAC einen Benutzer für die IAM-Authentifizierung zuweisen

Es gibt mehrere Möglichkeiten, attributbasierte Zugriffskontrolle (ABAC) für ein Studio zu konfigurieren. Sie können beispielsweise ein oder mehrere Tags an ein EMR Studio anhängen

und dann eine IAM-Richtlinie erstellen, die die `CreateStudioPresignedUrl`-Aktion auf ein bestimmtes Studio oder eine Gruppe von Studios mit diesen Tags beschränkt.

Sie können Tags während oder nach der Erstellung von Studio hinzufügen. Verwenden Sie den Befehl [AWS CLI `emr add-tags`](#), um Tags zu einem bestehenden Studio hinzuzufügen. Das folgende Beispiel fügt einem EMR Studio ein Tag mit dem Schlüssel-Wert-Paar `Team = Data Analytics` hinzu.

```
aws emr add-tags --resource-id <example-studio-id> --tags Team="Data Analytics"
```

Die folgende Beispiel-Berechtigungsrichtlinie ermöglicht die `CreateStudioPresignedUrl`-Aktion für EMR Studios mit dem Tag key-value pair `Team = DataAnalytics`. Weitere Informationen zur Verwendung von Tags zur Zugriffssteuerung finden Sie unter [Zugriffskontrolle für Benutzer und Rollen mithilfe von Tags](#) oder [Zugriffskontrolle auf AWS -Ressourcen mithilfe von Tags](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateStudioPresignedUrl",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:CreateStudioPresignedUrl"
      ],
      "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/Team": "Data Analytics"
        }
      }
    }
  ]
}
```

Example Weisen Sie einem Studio mithilfe des `SourceIdentity` globalen Bedingungsschlüssels `aws:` einen Benutzer zu

Wenn Sie den IAM-Verbund verwenden, können Sie den globalen Bedingungsschlüssel `aws:SourceIdentity` in einer Berechtigungsrichtlinie verwenden, um Benutzern Studio-Zugriff zu gewähren, wenn sie Ihre IAM-Rolle für den Verbund übernehmen.

Sie müssen zunächst Ihren Identitätsanbieter (IdP) so konfigurieren, dass er eine identifizierende Zeichenfolge zurückgibt, z. B. eine E-Mail-Adresse oder einen Benutzernamen, wenn sich ein Benutzer authentifiziert und Ihre IAM-Rolle für den Verbund übernimmt. IAM setzt den globalen Bedingungsschlüssel `aws:SourceIdentity` auf die identifizierende Zeichenfolge, die von Ihrem IdP zurückgegeben wurde.

Weitere Informationen finden Sie im Blogbeitrag [Wie man die IAM-Rollenaktivität mit der Unternehmensidentität in Beziehung](#) setzt im AWS Sicherheits-Blog und in der Referenz [aws:](#) in `SourceIdentity` der Referenz zu den globalen Bedingungsschlüsseln.

Die folgende Beispielrichtlinie ermöglicht die `CreateStudioPresignedUrl` Aktion und gewährt Benutzern mit einer `aws:SourceIdentity`, die dem von `<example-studio-arn>` angegebenen `<example-source-identity>` Zugriff auf das EMR Studio entspricht.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticmapreduce:CreateStudioPresignedUrl",
      "Resource": "<example-studio-arn>",
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": "<example-source-identity>"
        }
      }
    }
  ]
}
```

IAM Identity Center

Wenn Sie einem EMR Studio einen Benutzer oder eine Gruppe zuweisen, geben Sie eine Sitzungsrichtlinie an, die detaillierte Berechtigungen für diesen Benutzer oder diese Gruppe definiert, z. B. die Möglichkeit, einen neuen EMR-Cluster zu erstellen. Amazon EMR speichert diese Zuordnungen von Sitzungsrichtlinien. Sie können die Sitzungsrichtlinie eines Benutzers oder einer Gruppe nach der Zuweisung aktualisieren.

 Note

Die endgültigen Berechtigungen für einen Benutzer oder eine Gruppe sind eine Schnittmenge der in Ihrer EMR-Studio-Benutzerrolle definierten Berechtigungen und den in der Sitzungsrichtlinie für diesen Benutzer oder diese Gruppe definierten Berechtigungen. Wenn ein Benutzer zu mehr als einer Gruppe gehört, die dem Studio zugewiesen ist, verwendet EMR Studio eine Kombination von Berechtigungen für diesen Benutzer.

So weisen Sie einem EMR Studio mithilfe der Amazon-EMR-Konsole Benutzer oder Gruppen zu

1. Navigieren Sie zur neuen Amazon-EMR-Konsole und wählen Sie in der Seitennavigation die Option [Zur alten Konsole wechseln](#) aus. Weitere Informationen darüber, was Sie erwartet, wenn Sie zur alten Konsole wechseln, finden Sie unter [Verwenden der alten Konsole](#).
2. Wählen Sie in der linken Navigationsleiste EMR Studio aus.
3. Wählen Sie Ihren Studio-Namen aus der Studio-Liste oder wählen Sie das Studio aus und klicken Sie auf Details anzeigen, um die Studio-Detailseite zu öffnen.
4. Wählen Sie die Benutzer hinzufügen um die Suchtabelle Benutzer und Gruppen anzuzeigen.
5. Wählen Sie die Registerkarte Benutzer oder die Registerkarte Gruppen und geben Sie einen Suchbegriff in die Suchleiste ein, um einen Benutzer oder eine Gruppe zu finden.
6. Wählen Sie einen oder mehrere Benutzer oder Gruppen aus der Suchergebnisliste aus. Sie können zwischen der Registerkarte Benutzer und der Registerkarte Gruppen hin- und herwechseln.
7. Nachdem Sie Benutzer und Gruppen ausgewählt haben, die Sie dem Studio hinzufügen möchten, wählen Sie Hinzufügen. Sie sollten sehen, dass die Benutzer und Gruppen in der Studio-Benutzerliste angezeigt werden. Es kann einige Sekunden dauern, bis die Liste aktualisiert wird.
8. Folgen Sie den Anweisungen unter [Aktualisieren Sie die Berechtigungen für einen Benutzer oder eine Gruppe, die einem Studio zugewiesen ist](#), um die Studio-Berechtigungen für einen Benutzer oder eine Gruppe zu verfeinern.

Wie Sie EMR Studio einen Benutzer oder eine Gruppe mit AWS CLI zuweisen

Fügen Sie Ihre eigenen Werte für die folgenden `create-studio-session-mapping`-Argumente ein. Weitere Informationen über den Befehl `create-studio-session-mapping` finden Sie unter [AWS CLI -Befehlsreferenz](#).

- **--studio-id** – Die ID des Studios, dem Sie den Benutzer oder die Gruppe zuweisen möchten. Anleitungen zum Abrufen einer Studio-ID finden Sie unter [Studio-Details anzeigen](#).
- **--identity-name** – Der Name des Benutzers oder der Gruppe aus dem Identitätsspeicher. Weitere Informationen finden Sie unter [UserName](#) für Benutzer und [DisplayName](#) für Gruppen in der Identity Store-API-Referenz.
- **--identity-type** – Verwenden Sie entweder `USER` oder `GROUP`, um den Identitätstyp anzugeben.
- **--session-policy-arn** – Der Amazon-Ressourcenname (ARN) für die Sitzungsrichtlinie, die Sie mit dem Benutzer oder der Gruppe verknüpfen möchten. Beispiel, **arn:aws:iam::<aws-account-id>:policy/EMRStudio_Advanced_User_Policy**. Weitere Informationen finden Sie unter [Berechtigungsrichtlinien für EMR-Studio-Benutzer erstellen](#).

```
aws emr create-studio-session-mapping \  
  --studio-id <example-studio-id> \  
  --identity-name <example-identity-name> \  
  --identity-type <USER-or-GROUP> \  
  --session-policy-arn <example-session-policy-arn>
```

Note

Linux-Zeilenfortsetzungszeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

Verwenden Sie den `get-studio-session-mapping`-Befehl, um die neue Zuweisung zu überprüfen. **<example-identity-name>** Ersetzen Sie es durch den IAM Identity Center-Namen des Benutzers oder der Gruppe, den Sie aktualisiert haben.

```
aws emr get-studio-session-mapping \  
  --studio-id <example-studio-id> \  
  --identity-type <USER-or-GROUP> \
```

```
--identity-name <user-or-group-name> \
```

Aktualisieren Sie die Berechtigungen für einen Benutzer oder eine Gruppe, die einem Studio zugewiesen ist

IAM

Um Benutzer- oder Gruppenberechtigungen zu aktualisieren, wenn Sie den IAM-Authentifizierungsmodus verwenden, verwenden Sie IAM, um die IAM-Berechtigungsrichtlinien zu ändern, die Ihren IAM-Identitäten (Benutzern, Gruppen oder Rollen) zugeordnet sind.

Weitere Informationen finden Sie unter [Benutzerberechtigungen für den IAM-Authentifizierungsmodus](#).

IAM Identity Center

So aktualisieren Sie die EMR-Studio-Berechtigungen für einen Benutzer oder eine Gruppe mithilfe der Konsole

1. Navigieren Sie zur neuen Amazon-EMR-Konsole und wählen Sie in der Seitennavigation die Option Zur alten Konsole wechseln aus. Weitere Informationen darüber, was Sie erwartet, wenn Sie zur alten Konsole wechseln, finden Sie unter [Verwenden der alten Konsole](#).
2. Wählen Sie in der linken Navigationsleiste EMR Studio aus.
3. Wählen Sie Ihren Studio-Namen aus der Studio-Liste oder wählen Sie das Studio aus und klicken Sie auf Details anzeigen, um die Studio-Detailseite zu öffnen.
4. Suchen Sie in der Studio-Benutzerliste auf der Studio-Detailseite nach dem Benutzer oder der Gruppe, die Sie aktualisieren möchten. Sie können nach Namen oder Identitätstyp suchen.
5. Wählen Sie den Benutzer oder die Gruppe aus, den bzw. die Sie aktualisieren möchten, und wählen Sie Richtlinie zuweisen, um das Dialogfeld Sitzungsrichtlinie zu öffnen.
6. Wählen Sie eine Richtlinie aus, die auf den Benutzer oder die Gruppe angewendet werden soll, den Sie in Schritt 5 ausgewählt haben, und klicken Sie dann auf Richtlinie anwenden. In der Liste Studio-Benutzer sollte der Richtlinienname in der Spalte Sitzungsrichtlinie für den Benutzer oder die Gruppe angezeigt werden, den Sie aktualisiert haben.

So aktualisieren Sie die EMR Studio-Berechtigungen für einen Benutzer oder eine Gruppe mithilfe der AWS CLI

Fügen Sie Ihre eigenen Werte für die folgenden `update-studio-session-mappings`-Argumente ein. Weitere Informationen über den Befehl `update-studio-session-mappings` finden Sie unter [AWS CLI -Befehlsreferenz](#).

```
aws emr update-studio-session-mapping \  
  --studio-id <example-studio-id> \  
  --identity-name <name-of-user-or-group-to-update> \  
  --session-policy-arn <new-session-policy-arn-to-apply> \  
  --identity-type <USER-or-GROUP> \  
  --session-policy-arn <new-session-policy-arn-to-apply> \  
  --identity-type <USER-or-GROUP> \
```

Verwenden Sie den `get-studio-session-mapping`-Befehl, um die neue Zuweisung der Sitzungsrichtlinie zu überprüfen. `<example-identity-name>` Ersetzen Sie es durch den IAM Identity Center-Namen des Benutzers oder der Gruppe, den Sie aktualisiert haben.

```
aws emr get-studio-session-mapping \  
  --studio-id <example-studio-id> \  
  --identity-type <USER-or-GROUP> \  
  --identity-name <user-or-group-name> \
```

Einen Benutzer oder eine Gruppe aus einem Studio entfernen

IAM

Um einen Benutzer oder eine Gruppe aus einem EMR Studio zu entfernen, wenn Sie den IAM-Authentifizierungsmodus verwenden, müssen Sie dem Benutzer den Zugriff auf das Studio entziehen, indem Sie die IAM-Berechtigungsrichtlinie des Benutzers neu konfigurieren.

Gehen Sie in der folgenden Beispielrichtlinie davon aus, dass Sie über ein EMR Studio mit dem Tag-Schlüsselwertpaar `Team = Quality Assurance` verfügen. Gemäß der Richtlinie kann der Benutzer auf Studios zugreifen, die mit dem `Team`-Schlüssel gekennzeichnet sind, dessen Wert entweder `Data Analytics` oder `Quality Assurance` entspricht. Um den Benutzer aus dem Studio zu entfernen, das mit `Team = Quality Assurance` markiert ist, entfernen Sie `Quality Assurance` aus der Liste der Tag-Werte.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowCreateStudioPresignedUrl",  
      "Effect": "Allow",  
      "Principal": "AWS",  
      "Action": "emr:CreateStudioPresignedUrl",  
      "Resource": "arn:aws:emr:*:*:studio/*",  
      "Condition": {"  
        "StringEquals": {"  
          "emr:TagKeys": "Team",  
          "emr:TagValues": "Quality Assurance",  
          "emr:TagValueEquals": "Quality Assurance"  
        }  
      }  
    }  
  ]  
}
```

```

    "Effect": "Allow",
    "Action": [
        "elasticmapreduce:CreateStudioPresignedUrl"
    ],
    "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
    "Condition": {
        "StringEquals": {
            "emr:ResourceTag/Team": [
                "Data Analytics",
                "Quality Assurance"
            ]
        }
    }
}

```

IAM Identity Center

So entfernen Sie einen Benutzer oder eine Gruppe mithilfe der Konsole aus einem EMR Studio

1. Navigieren Sie zur neuen Amazon-EMR-Konsole und wählen Sie in der Seitennavigation die Option Zur alten Konsole wechseln aus. Weitere Informationen darüber, was Sie erwartet, wenn Sie zur alten Konsole wechseln, finden Sie unter [Verwenden der alten Konsole](#).
2. Wählen Sie in der linken Navigationsleiste EMR Studio aus.
3. Wählen Sie Ihren Studio-Namen aus der Studio-Liste oder wählen Sie das Studio aus und klicken Sie auf Details anzeigen, um die Studio-Detailseite zu öffnen.
4. Suchen Sie in der Liste Studio-Benutzer auf der Studio-Detailseite nach dem Benutzer oder der Gruppe, die Sie aus dem Studio entfernen möchten. Sie können nach Namen oder Identitätstyp suchen.
5. Wählen Sie den Benutzer oder die Gruppe aus, die Sie löschen möchten, wählen Sie Löschen und bestätigen Sie das Löschen. Der Benutzer oder die Gruppe, den Sie gelöscht haben, verschwindet aus der Liste Studio-Benutzer.

Wie Sie einen Benutzer oder eine Gruppe aus einem EMR Studio mit der AWS CLI entfernen

Fügen Sie Ihre eigenen Werte für die folgenden `delete-studio-session-mapping`-Argumente ein. Weitere Informationen über den Befehl `delete-studio-session-mapping` finden Sie unter [AWS CLI -Befehlsreferenz](#).

```
aws emr delete-studio-session-mapping \  
--studio-id <example-studio-id> \  
--identity-type <USER-or-GROUP> \  
--identity-name <name-of-user-or-group-to-delete> \
```

Amazon EMR Studio-Ressourcen überwachen, aktualisieren und löschen

Dieser Abschnitt enthält Anweisungen, die Ihnen helfen, eine EMR-Studio-Ressource zu überwachen, zu aktualisieren oder zu löschen. Informationen zum Zuweisen von Benutzern oder zum Aktualisieren von Benutzerberechtigungen finden Sie unter [EMR-Studio-Benutzer zuweisen und verwalten](#).

Studio-Details anzeigen

Console

So zeigen Sie Details zu einem EMR Studio mit der neuen Konsole an

1. Öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie im linken Navigationsbereich unter EMR Studio die Option Studios aus.
3. Wählen Sie das Studio aus der Studio-Liste aus, um die Studio-Detailseite zu öffnen. Die Studio-Detailseite enthält Informationen zu den Studio-Einstellungen, z. B. die Studio-Beschreibung, VPC und Subnetze.

CLI

Um Details für ein EMR Studio anhand der Studio-ID abzurufen, verwenden Sie AWS CLI

Verwenden Sie den folgenden `describe-studio` AWS CLI Befehl, um detaillierte Informationen zu einem bestimmten EMR Studio abzurufen. Weitere Informationen finden Sie in der [AWS CLI - Befehlsreferenz](#).

```
aws emr describe-studio \  
--studio-id <id-of-studio-to-describe> \
```

Um eine Liste von EMR Studios abzurufen, verwenden Sie AWS CLI

Verwenden Sie den folgenden `list-studios` AWS CLI -Befehl. Weitere Informationen finden Sie in der [AWS CLI -Befehlsreferenz](#).

```
aws emr list-studios
```

Es folgt ein Beispiel für einen Rückgabewert für den `list-studios`-Befehl im JSON-Format.

```
{
  "Studios": [
    {
      "AuthMode": "IAM",
      "VpcId": "vpc-b21XXXXX",
      "Name": "example-studio-name",
      "Url": "https://es-7HWP74SNGDXXXXXXXXXXXXXXXXX.emrstudio-prod.us-east-1.amazonaws.com",
      "CreationTime": 1605672582.781,
      "StudioId": "es-7HWP74SNGDXXXXXXXXXXXXXXXXX",
      "Description": "example studio description"
    }
  ]
}
```

Amazon-EMR-Studio-Aktionen überwachen

EMR-Studio- und API-Aktivitäten anzeigen

EMR Studio ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer IAM-Rolle oder einem anderen AWS Dienst in EMR Studio ausgeführt wurden. CloudTrail erfasst API-Aufrufe für EMR Studio als Ereignisse. Sie können Ereignisse in der CloudTrail Konsole unter <https://console.aws.amazon.com/cloudtrail/> anzeigen.

EMR-Studio-Ereignisse liefern Informationen darüber, welcher Studio- oder IAM-Benutzer eine Anfrage stellt und um welche Art von Anfrage es sich handelt.

Note

Clusterinterne Aktionen wie das Ausführen von Notebook-Aufträgen werden AWS CloudTrail nicht ausgegeben.

Sie können auch einen Trail für die kontinuierliche Bereitstellung von EMR CloudTrail Studio-Ereignissen an einen Amazon S3 S3-Bucket erstellen. Weitere Informationen finden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

CloudTrail Beispielereignis: Ein Benutzer ruft die API auf DescribeStudio

Das Folgende ist ein AWS CloudTrail Beispielereignis, das erzeugt wird, wenn ein Benutzer,admin, die [DescribeStudio](#)API aufruft. CloudTrail zeichnet den Benutzernamen auf alsadmin.

Note

Um Studio-Details zu schützen, DescribeStudio schließt das EMR Studio API-Ereignis für einen Wert für aus. responseElements

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDXXXXXXXXXXXXXXXXXXXX",
    "arn": "arn:aws:iam::653XXXXXXXXX:user/admin",
    "accountId": "653XXXXXXXXX",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "admin"
  },
  "eventTime": "2021-01-07T19:13:58Z",
  "eventSource": "elasticmapreduce.amazonaws.com",
  "eventName": "DescribeStudio",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.XX.XXX.XX",
  "userAgent": "aws-cli/1.18.188 Python/3.8.5 Darwin/18.7.0 botocore/1.19.28",
  "requestParameters": {
    "studioId": "es-905XXXXXXXXXXXXXXXXXXXX"
  },
  "responseElements": null,
  "requestID": "0fxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "eventID": "b0xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "653XXXXXXXXX"
}
```

```
}
```

Spark-Benutzer- und Jobaktivitäten anzeigen

Um die Spark-Jobaktivitäten von Amazon-EMR-Studio-Benutzern anzuzeigen, können Sie den Benutzerwechsel in einem Cluster konfigurieren. Beim Identitätswechsel wird jeder Spark-Job, der von einem Workspace aus eingereicht wird, dem Studio-Benutzer zugeordnet, der den Code ausgeführt hat.

Wenn der Benutzerwechsel aktiviert ist, erstellt Amazon EMR ein HDFS-Benutzerverzeichnis auf dem Primärknoten des Clusters für jeden Benutzer, der Code im Workspace ausführt. Wenn beispielsweise ein Benutzer `studio-user-1@example.com` Code ausführt, können Sie eine Verbindung zum Primärknoten herstellen und sehen, dass `hadoop fs -ls /user` ein Verzeichnis für `studio-user-1@example.com` hat.

Um den Spark-Benutzerwechsel einzurichten, legen Sie die folgenden Eigenschaften in den folgenden Konfigurationsklassifizierungen fest:

- `core-site`
- `livy-conf`

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "hadoop.proxyuser.livy.groups": "*",
      "hadoop.proxyuser.livy.hosts": "*"
    }
  },
  {
    "Classification": "livy-conf",
    "Properties": {
      "livy.impersonation.enabled": "true"
    }
  }
]
```

Informationen zum Anzeigen von Verlaufsserverseiten finden Sie unter [Debuggen von Anwendungen und Aufträgen mit EMR Studio](#). Sie können auch mithilfe von SSH eine Verbindung zum Primärknoten des Clusters herstellen, um die Webschnittstellen der Anwendungen anzuzeigen.

Weitere Informationen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Ein Amazon EMR Studio aktualisieren

Nachdem Sie ein EMR Studio erstellt haben, können Sie die folgenden Attribute mit dem AWS CLI aktualisieren:

- Name
- Beschreibung
- Standard-S3-Speicherort
- Subnetze

Um ein EMR Studio mit dem zu aktualisieren AWS CLI

Verwenden Sie den `update-studio` AWS CLI Befehl, um ein EMR Studio zu aktualisieren. Weitere Informationen finden Sie in der [AWS CLI -Befehlsreferenz](#).

Note

Sie können ein Studio mit maximal 5 Subnetzen verknüpfen. Diese Subnetze müssen zur gleichen VPC gehören wie Studio. Die Liste der Subnetze IDs , die Sie an den `update-studio` Befehl senden, kann ein neues Subnetz enthalten IDs, muss aber auch das gesamte Subnetz enthalten, IDs das Sie dem Studio bereits zugeordnet haben. Sie können keine Subnetze aus einem Studio entfernen.

```
aws emr update-studio \  
  --studio-id <example-studio-id-to-update> \  
  --name <example-new-studio-name> \  
  --subnet-ids <old-subnet-id-1 old-subnet-id-2 old-subnet-id-3 new-subnet-id> \  

```

Verwenden Sie den `describe-studio` AWS CLI Befehl und geben Sie Ihre Studio-ID an, um die Änderungen zu überprüfen. Weitere Informationen finden Sie in der [AWS CLI -Befehlsreferenz](#).

```
aws emr describe-studio \  
  --studio-id <id-of-updated-studio> \  

```

Löschen Sie ein Amazon EMR Studio und Workspaces

Wenn Sie ein Studio löschen, löscht EMR Studio alle IAM-Identity-Center-Benutzer- und Gruppenzuweisungen, die dem Studio zugeordnet sind.

Note

Wenn Sie ein Studio löschen, löscht Amazon EMR die mit diesem Studio verknüpften Workspaces nicht. Sie müssen die Workspaces in Ihrem Studio separat löschen.

WorkSpaces löschen

Console

Da es sich bei jedem EMR-Studio-Workspace um eine EMR-Notebook-Instance handelt, können Sie die Verwaltungskonsolle für Amazon EMR verwenden, um Workspaces zu löschen. Sie können Workspaces mit der Amazon-EMR-Konsole löschen, bevor oder nachdem Sie Ihr Studio gelöscht haben.

Um einen Workspace mit der Amazon-EMR-Konsole zu löschen

1. Navigieren Sie zur neuen Amazon-EMR-Konsole und wählen Sie in der Seitennavigation die Option [Zur alten Konsole wechseln](#) aus. Weitere Informationen darüber, was Sie erwartet, wenn Sie zur alten Konsole wechseln, finden Sie unter [Verwenden der alten Konsole](#).
2. Wählen Sie Notebooks.
3. Wählen Sie die Arbeitsbereiche aus, die Sie löschen möchten.
4. Wählen Sie Löschen und nochmals Löschen aus, um den Vorgang zu bestätigen.
5. Folgen Sie den Anweisungen zum [Löschen von Objekten](#) im Konsolen-Benutzerhandbuch für Amazon Simple Storage Service, um die mit dem gelöschten Workspace verknüpften Notebookdateien aus Amazon S3 zu entfernen.

EMR Studio UI

From the Workspace UI

Löschen Sie einen Workspace und die zugehörigen Backup-Dateien aus EMR Studio

1. Melden Sie sich mit Ihrer Studio-Zugriffs-URL bei Ihrem EMR Studio an und wählen Sie in der linken Navigationsleiste Workspaces aus.
2. Suchen Sie in der Liste nach Ihrem Workspace und aktivieren Sie das Kontrollkästchen neben dessen Namen. Sie können mehrere Arbeitsbereiche zum gleichzeitigen Löschen auswählen.
3. Wählen Sie oben rechts in der Liste der Arbeitsbereiche die Option Löschen aus und bestätigen Sie, dass Sie die ausgewählten Arbeitsbereiche löschen möchten. Wählen Sie zur Bestätigung Delete.
4. Wenn Sie die Notebookdateien, die mit dem gelöschten Workspace verknüpft waren, aus Amazon S3 entfernen möchten, folgen Sie den Anweisungen zum [Löschen von Objekten](#) im Konsole-Benutzerhandbuch von Amazon Simple Storage Service. Wenn Sie das Studio nicht erstellt haben, wenden Sie sich an Ihren Studio-Administrator, um den Amazon-S3-Backup-Speicherort für den gelöschten Workspace zu ermitteln.

From the Workspaces list

Löschen Sie einen Workspace und die zugehörigen Backup-Dateien aus der Workspaces-Liste

1. Navigieren Sie in der Konsole zur Workspace-Liste.
2. Wählen Sie den Workspace, den Sie löschen möchten, aus der Liste aus und klicken Sie dann auf Aktionen.
3. Wählen Sie Löschen.
4. Wenn Sie die Notebookdateien, die mit dem gelöschten Workspace verknüpft waren, aus Amazon S3 entfernen möchten, folgen Sie den Anweisungen zum [Löschen von Objekten](#) im Konsole-Benutzerhandbuch von Amazon Simple Storage Service. Wenn Sie das Studio nicht erstellt haben, wenden Sie sich an Ihren Studio-Administrator, um den Amazon-S3-Backup-Speicherort für den gelöschten Workspace zu ermitteln.

Ein EMR Studio löschen

Console

So löschen Sie ein EMR Studio mit der neuen Konsole

1. Öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie im linken Navigationsbereich unter EMR Studio die Option Studios aus.
3. Wählen Sie das Studio aus der Studio-Liste mit dem Schalter links neben dem Studio-Namen aus. Wählen Sie Löschen.

Old console

So löschen Sie ein EMR Studio mit der alten Konsole

1. Öffnen Sie die Amazon EMR-Konsole zu <https://console.aws.amazon.com/elasticmapreduce/Hause>.
2. Wählen Sie in der linken Navigationsleiste EMR Studio aus.
3. Wählen Sie das Studio aus der Studio-Liste aus und klicken Sie auf Löschen.

CLI

Um ein EMR Studio mit dem zu löschen AWS CLI

Verwenden Sie den `delete-studio` AWS CLI Befehl, um ein EMR Studio zu löschen. Weitere Informationen finden Sie in der [AWS CLI -Befehlsreferenz](#).

```
aws emr delete-studio --studio-id <id-of-studio-to-delete>
```

Verschlüsselung von EMR Studio Workspace-Notizbüchern und -Dateien

In EMR Studio können Sie verschiedene Arbeitsbereiche erstellen und konfigurieren, um Notizbücher zu organisieren und auszuführen. In diesen Arbeitsbereichen werden Notizbücher und zugehörige Dateien in Ihrem angegebenen Amazon S3 S3-Bucket gespeichert. Standardmäßig werden diese Dateien mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) mit serverseitiger Verschlüsselung als Basisverschlüsselungsebene verschlüsselt. Sie können sich auch dafür entscheiden, Ihre Dateien mit kundenverwalteten KMS-Schlüsseln (SSE-KMS) zu verschlüsseln. Sie können dies tun, indem Sie die Amazon EMR-Managementkonsole oder das AWS CLI AWS UND-SDK verwenden, wenn Sie ein EMR Studio erstellen.

Die EMR Studio-Workspace-Speicherverschlüsselung ist in allen [Regionen](#) verfügbar, in denen EMR Studio verfügbar ist.

Voraussetzungen

Bevor Sie das EMR Studio-Workspace-Notizbuch und die Dateien verschlüsseln können, müssen Sie [einen symmetrischen Kundenmanager-Schlüssel \(CMK\) in derselben AWS-Konto Region wie Ihr EMR Studio erstellen](#). AWS Key Management Service

Ihre Ressourcenrichtlinie AWS KMS muss über die erforderlichen Zugriffsberechtigungen für die Servicerolle Ihres EMR Studio verfügen. Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die Mindestzugriffsberechtigungen für die EMR Studio Workspace-Speicherverschlüsselung gewährt:

```
{
  "Sid": "AllowEMRStudioServiceRoleAccess",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<ACCOUNT_ID>:role/<ROLE_NAME>"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:ReEncryptFrom",
    "kms:ReEncryptTo",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "<ACCOUNT_ID>",
      "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::<S3_BUCKET_NAME>",
      "kms:ViaService": "s3.<AWS_REGION>.amazonaws.com"
    }
  }
}
```

Ihre EMR Studio-Servicerolle muss auch über die Zugriffsberechtigungen verfügen, um Ihren AWS KMS Schlüssel verwenden zu können. Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die die Mindestzugriffsberechtigungen für die EMR Studio Workspace-Speicherverschlüsselung gewährt:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowEMRStudioWorkspaceStorageEncryptionAccess",
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey",
      "kms:ReEncryptFrom",
      "kms:ReEncryptTo",
      "kms:DescribeKey"
    ],
    "Resource": ["arn:aws:kms:<REGION>:<ACCOUNT_ID>:key/<KEY_IDENTITYIFIER>"]
  }
]
```

Neues EMR Studio erstellen

Gehen Sie wie folgt vor, um ein neues EMR Studio zu erstellen, das Workspace Storage Encryption verwendet.

1. Öffnen Sie die Amazon-EMR-Konsole unter <https://console.aws.amazon.com/elasticmapreduce/>.
2. Wählen Sie Studios und anschließend Create Studio aus.
3. Geben Sie für S3-Speicherort einen Amazon S3 S3-Pfad ein oder wählen Sie ihn aus. Dies ist der Amazon S3 S3-Speicherort, an dem Amazon EMR Workspace-Notizbücher und -Dateien speichert.
4. Geben Sie für die Servicerolle eine IAM-Rolle ein oder wählen Sie sie aus. Dies ist die IAM-Rolle, die Amazon EMR übernimmt.
5. Wählen Sie Workspace-Dateien mit Ihrem eigenen Schlüssel verschlüsseln. AWS KMS
6. Geben Sie einen AWS KMS Schlüssel ein, der zum Verschlüsseln von Workspace-Notizbüchern und -Dateien in Amazon S3 verwendet werden soll, oder wählen Sie einen aus.
7. Wählen Sie Create Studio oder Create Studio and Launch Workspaces.
8. Wählen Sie Workspace-Dateien mit Ihrem eigenen AWS KMS Schlüssel verschlüsseln.
9. Geben Sie eine ein AWS KMS , die zum Verschlüsseln von Workspace-Notizbüchern und -Dateien in Amazon S3 verwendet werden soll, oder wählen Sie eine aus.
10. Wählen Sie Save Changes.

Die folgenden Schritte zeigen, wie Sie ein EMR Studio aktualisieren und die Workspace-Speicherverschlüsselung einrichten.

1. Öffnen Sie die Amazon-EMR-Konsole unter <https://console.aws.amazon.com/elasticmapreduce/>.
2. Wählen Sie ein vorhandenes EMR Studio aus der Liste aus und klicken Sie dann auf Bearbeiten.
3. Wählen Sie Workspace-Dateien mit Ihrem eigenen AWS KMS Schlüssel verschlüsseln.
4. Geben Sie eine ein AWS KMS , die zum Verschlüsseln von Workspace-Notizbüchern und -Dateien in Amazon S3 verwendet werden soll, oder wählen Sie eine aus.
5. Wählen Sie Save Changes.

Definieren Sie Sicherheitsgruppen zur Steuerung des Netzwerkverkehrs in EMR Studio

Informationen über die EMR-Studio-Sicherheitsgruppen

Amazon EMR Studio verwendet zwei Sicherheitsgruppen, um den Netzwerkverkehr zwischen Workspaces im Studio und einem angeschlossenen Amazon EMR-Cluster, der auf Amazon läuft, zu kontrollieren: EC2

- Eine Engine-Sicherheitsgruppe, die Port 18888 verwendet, um mit einem verbundenen Amazon EMR-Cluster zu kommunizieren, der auf Amazon läuft. EC2
- Eine Workspace-Sicherheitsgruppe, die den Workspaces in einem Studio zugeordnet ist. Diese Sicherheitsgruppe umfasst eine ausgehende HTTPS-Regel, die es dem Workspace ermöglicht, Datenverkehr ins Internet weiterzuleiten, und muss ausgehenden Datenverkehr ins Internet an Port 443 zulassen, um die Verknüpfung von Git-Repositorys mit einem Workspace zu ermöglichen.

EMR Studio verwendet diese Sicherheitsgruppen zusätzlich zu allen Sicherheitsgruppen, die einem an einen Workspace angeschlossenen EMR-Cluster zugeordnet sind.

Sie müssen diese Sicherheitsgruppen erstellen, wenn Sie das verwenden, um ein AWS CLI Studio zu erstellen.

Note

Sie können die Sicherheitsgruppen für EMR Studio mit Regeln anpassen, die auf Ihre Umgebung zugeschnitten sind. Sie müssen jedoch die auf dieser Seite aufgeführten Regeln

einbeziehen. Ihre Workspace-Sicherheitsgruppe kann keinen eingehenden Datenverkehr zulassen, und die Engine-Sicherheitsgruppe muss eingehenden Datenverkehr von der Workspace-Sicherheitsgruppe zulassen.

Die standardmäßigen EMR-Studio-Sicherheitsgruppen verwenden

Wenn Sie die Amazon-EMR-Konsole verwenden, können Sie die folgenden Standardsicherheitsgruppen auswählen. Die Standardsicherheitsgruppen werden von EMR Studio in Ihrem Namen erstellt und enthalten die Mindestregeln für eingehende und ausgehende Nachrichten für Workspaces in einem EMR Studio.

- `DefaultEngineSecurityGroup`
- `DefaultWorkspaceSecurityGroupGit` oder `DefaultWorkspaceSecurityGroupWithoutGit`

Voraussetzungen

Um die Sicherheitsgruppen für EMR Studio zu erstellen, benötigen Sie eine Amazon Virtual Private Cloud (VPC) für das Studio. Sie wählen diese VPC aus, wenn Sie die Sicherheitsgruppen erstellen. Dies sollte die gleiche VPC sein, die Sie beim Erstellen des Studios angeben. Wenn Sie Amazon EMR in EKS mit EMR Studio verwenden möchten, wählen Sie die VPC für Ihre Amazon-EKS-Cluster-Workerknoten aus.

Anweisungen

Folgen Sie den Anweisungen unter [Erstellen einer Sicherheitsgruppe](#) im EC2 Amazon-Benutzerhandbuch für Linux-Instances, um eine Engine-Sicherheitsgruppe und eine Workspace-Sicherheitsgruppe in Ihrer VPC zu erstellen. Die Sicherheitsgruppen müssen die in den folgenden Tabellen zusammengefassten Regeln enthalten.

Wenn Sie Sicherheitsgruppen für EMR Studio erstellen, beachten Sie die IDs für beide. Sie geben jede Sicherheitsgruppe anhand ihrer ID an, wenn Sie ein Studio erstellen.

Engine-Sicherheitsgruppe

EMR Studio verwendet Port 18888 für die Kommunikation mit einem angeschlossenen Cluster.

Regeln für eingehenden Datenverkehr

Typ	Protokoll	Port	Bestimmungsort	Beschreibung
TCP	TCP	18888	Ihre EMR Studio Workspace-Sicherheitsgruppe.	Lassen Sie Datenverkehr von allen Ressourcen in der Workspace-Sicherheitsgruppe für EMR Studio zu.

WorkSpaces-Sicherheitsgruppe

Diese Sicherheitsgruppe ist den Workspaces in einem EMR Studio zugeordnet.

Regeln für ausgehenden Datenverkehr

Typ	Protokoll	Port	Bestimmungsort	Beschreibung
TCP	TCP	18888	Ihre EMR-Studio-Engine-Sicherheitsgruppe.	Erlauben Sie Datenverkehr zu allen Ressourcen in der Engine-Sicherheitsgruppe für EMR Studio.
HTTPS	TCP	443	0.0.0.0/0	Erlaube Datenverkehr im Internet, um öffentlich gehostete Git-Repositorys mit Workspaces zu verknüpfen.

AWS CloudFormation Vorlagen für Amazon EMR Studio erstellen

Informationen über die EMR-Studio-Clustervorlagen

Sie können AWS CloudFormation Vorlagen erstellen, um EMR Studio-Benutzern zu helfen, neue Amazon EMR-Cluster in einem Workspace zu starten. CloudFormation Vorlagen sind formatierte Textdateien in JSON oder YAML. In einer Vorlage beschreiben Sie einen Stapel von AWS Ressourcen und erklären, CloudFormation wie Sie diese Ressourcen für Sie bereitstellen können. Für EMR Studio können Sie eine oder mehrere Vorlagen erstellen, die einen Amazon-EMR-Cluster beschreiben.

Sie organisieren Ihre Vorlagen in AWS Service Catalog. AWS Service Catalog ermöglicht es Ihnen, häufig bereitgestellte IT-Services, sogenannte Produkte, zu erstellen und zu verwalten AWS. Sie sammeln Ihre Vorlagen als Produkte in einem Portfolio, das Sie mit Ihren EMR-Studio-Benutzern teilen. Nachdem Sie Cluster-Vorlagen erstellt haben, können Studio-Benutzer mit einer Ihrer Vorlagen einen neuen Cluster für einen Workspace starten. Benutzer müssen über die Berechtigung zum Erstellen neuer Cluster aus Vorlagen verfügen. Sie können Benutzerberechtigungen in Ihren [EMR-Studio-Berechtigungsrichtlinien festlegen](#).

Weitere Informationen zu CloudFormation Vorlagen finden Sie unter [Vorlagen](#) im AWS CloudFormation Benutzerhandbuch. Weitere Informationen zu AWS Service Catalog finden Sie unter [Was ist AWS Service Catalog](#).

Das folgende Video veranschaulicht, wie Sie Cluster-Vorlagen in AWS Service Catalog für EMR Studio einrichten. Weitere Informationen finden Sie auch im Blogbeitrag [Aufbau einer Self-Service-Umgebung für jeden Geschäftsbereich mithilfe von Amazon EMR und Service Catalog](#).

Optionale Vorlageparameter

Sie können zusätzliche Optionen in den [Parameters](#) Abschnitt Ihrer Vorlage aufnehmen. Mit Parametern können Studio-Benutzer benutzerdefinierte Werte für einen Cluster eingeben oder auswählen. Sie könnten beispielsweise einen Parameter hinzufügen, mit dem Benutzer eine bestimmte Amazon-EMR-Version auswählen können. Weitere Informationen finden Sie unter [Parameter](#) im AWS CloudFormation -Benutzerhandbuch.

Der folgende Parameters-Beispielabschnitt definiert zusätzliche Eingabeparameter wie `ClusterName`, `EmrRelease-Version` und `ClusterInstanceType`.

```
Parameters:
```

```
ClusterName:
  Type: "String"
  Default: "Cluster_Name_Placeholder"
EmrRelease:
  Type: "String"
  Default: "emr-6.2.0"
  AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
ClusterInstanceType:
  Type: "String"
  Default: "m5.xlarge"
  AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"
```

Wenn Sie Parameter hinzufügen, werden Studio-Benutzern nach der Auswahl einer Clustervorlage zusätzliche Formularoptionen angezeigt. Die folgende Abbildung zeigt zusätzliche Formularoptionen für `EmrReleaseVersion`, `ClusterName`, und `InstanceType`.

▼ Advanced configuration

To run your fully-managed Jupyter Notebook, you need to attach the Workspace to an EMR cluster. You can create a new cluster or

- ☐ Attach Workspace to an EMR cluster
Run your Workspace by choosing a cluster from a list of preset, running clusters.

- ☒ Use a cluster template
Provision a new EMR cluster from a pre-defined template.

Use a cluster template

Select from pre-defined cluster templates. When you choose "Create Workspace", a cluster will be created using the selected template

Cluster template

one-node-cluster ▼

Description:

one node cluster for bugbash

EmrRelease

emr-6.2.0 ▼

ClusterName

Cluster_Name_Placeholder

SubnetId

subnet-1643da37

InstanceType

m5.xlarge ▼

Voraussetzungen

Bevor Sie eine Clustervorlage erstellen, stellen Sie sicher, dass Sie über IAM-Berechtigungen für den Zugriff auf die Administratorkonsolenansicht von Service Catalog verfügen. Sie benötigen außerdem die erforderlichen IAM-Berechtigungen, um die administrativen Aufgaben von Service Catalog auszuführen. Weitere Informationen finden Sie unter [Service-Catalog-Administratoren Berechtigungen erteilen](#).

EMR-Cluster-Vorlagen erstellen

So erstellen Sie EMR-Clustervorlagen mithilfe von Service Catalog

1. Erstellen Sie eine oder mehrere CloudFormation Vorlagen. Wo Sie Ihre Vorlagen speichern, liegt bei Ihnen. Da es sich bei Vorlagen um formatierte Textdateien handelt, können Sie sie auf Amazon S3 hochladen oder in Ihrem lokalen Dateisystem speichern. Weitere Informationen zu CloudFormation Vorlagen finden Sie unter [Vorlagen](#) im AWS CloudFormation Benutzerhandbuch.

Verwenden Sie die folgenden Regeln, um Ihre Vorlagen zu benennen, oder vergleichen Sie Ihre Namen mit dem Muster `[a-zA-Z0-9][a-zA-Z0-9._-]*`.

- Vorlagennamen müssen mit einer Ziffer oder einem Buchstaben beginnen.
- Vorlagennamen dürfen nur aus Buchstaben, Ziffern, Punkten (.), Unterstrichen (_) und Bindestrichen (-) bestehen.

Jede Cluster-Vorlage, die Sie erstellen, muss die folgenden Optionen enthalten:

Eingabeparameter

- `ClusterName` — Ein Name für den Cluster, der Benutzern hilft, ihn nach der Bereitstellung zu identifizieren.

Ausgabe

- `ClusterId` – Die ID des neu bereitgestellten EMR-Clusters.

Im Folgenden finden Sie eine AWS CloudFormation Beispielvorlage im YAML-Format für einen Cluster mit zwei Knoten. Die Beispielvorlage enthält die erforderlichen Vorlagenoptionen und definiert zusätzliche Eingabeparameter für `EmrRelease` und `ClusterInstanceType`.

```
awsTemplateFormatVersion: 2010-09-09

Parameters:
  ClusterName:
    Type: "String"
    Default: "Example_Two_Node_Cluster"
  EmrRelease:
```

```

    Type: "String"
    Default: "emr-6.2.0"
    AllowedValues:
      - "emr-6.2.0"
      - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
    Default: "m5.xlarge"
    AllowedValues:
      - "m5.xlarge"
      - "m5.2xlarge"

Resources:
  EmrCluster:
    Type: AWS::EMR::Cluster
    Properties:
      Applications:
        - Name: Spark
        - Name: Livy
        - Name: JupyterEnterpriseGateway
        - Name: Hive
      EbsRootVolumeSize: '10'
      Name: !Ref ClusterName
      JobFlowRole: EMR_EC2_DefaultRole
      ServiceRole: EMR_DefaultRole_V2
      ReleaseLabel: !Ref EmrRelease
      VisibleToAllUsers: true
      LogUri:
        Fn::Sub: 's3://aws-logs-${AWS::AccountId}-${AWS::Region}/elasticmapreduce/'
      Instances:
        TerminationProtected: false
        Ec2SubnetId: 'subnet-ab12345c'
        MasterInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
        CoreInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
          Market: ON_DEMAND
          Name: Core

Outputs:
  ClusterId:
    Value:

```


Ref: EmrCluster
Description: The ID of the EMR cluster

2. Erstellen Sie ein Portfolio für Ihre Cluster-Vorlagen in demselben AWS Konto wie Ihr Studio.
 - a. Öffnen Sie die AWS Service Catalog Konsole unter <https://console.aws.amazon.com/servicecatalog/>.
 - b. Wählen Sie im linken Navigationsmenü Portfolios.
 - c. Geben Sie auf der Seite Portfolio erstellen die erforderlichen Informationen ein.
 - d. Wählen Sie Erstellen. AWS Service Catalog erstellt das Portfolio und zeigt die Portfoliodetails an.
3. Führen Sie die folgenden Schritte aus, um Ihre Cluster-Vorlagen als AWS Service Catalog -Produkte hinzuzufügen.
 - a. Navigieren Sie in der AWS Service Catalog -Managementkonsole unter Administration zur Seite Produkte.
 - b. Wählen Sie Neues Produkt hochladen.
 - c. Geben Sie einen Produktnamen und einen Eigentümer ein.
 - d. Geben Sie Ihre Vorlagendatei unter Versionsdetails an.
 - e. Wähle Überprüfen, um deine Produkteinstellungen zu überprüfen, und wähle dann Produkt erstellen.
4. Führen Sie die folgenden Schritte aus, um Ihre Produkte zu Ihrem Portfolio hinzuzufügen.
 - a. Navigieren Sie in der AWS Service Catalog -Managementkonsole unter Administration zur Seite Produkte.
 - b. Wählen Sie Ihr Produkt aus, klicken Sie auf Aktionen und anschließend auf Produkt zum Portfolio hinzufügen.
 - c. Wählen Sie Ihr Portfolio aus und klicken Sie dann auf Produkt zum Portfolio hinzufügen.
5. Legen Sie eine Beschränkung für die Markteinführung deiner Produkte fest. Eine Startbeschränkung ist eine IAM-Rolle, die Benutzerberechtigungen für die Markteinführung eines Produkts festlegt. Sie können Ihre Startbeschränkungen anpassen, müssen jedoch Berechtigungen zur Nutzung CloudFormation von Amazon EMR und AWS Service Catalog zulassen. Weitere Informationen und Anweisungen finden Sie unter [Startbeschränkungen für den Service Catalog](#).
6. Wenden Sie Ihre Markteinführungsbeschränkung auf jedes Produkt in Ihrem Portfolio an. Sie müssen die Markteinführungsbeschränkung auf jedes Produkt einzeln anwenden.

- a. Wählen Sie Ihr Portfolio auf der Portfolio-Seite in der AWS Service Catalog - Managementkonsole aus.
 - b. Wählen Sie die Registerkarte Constraints (Einschränkungen) und dann Create constraint (Einschränkung erstellen).
 - c. Wählen Sie Ihr Produkt aus und wählen Sie unter Einschränkungstyp die Option Starten aus. Klicken Sie auf Weiter.
 - d. Wählen Sie Ihre Startbeschränkungsrolle im Abschnitt Startbeschränkung aus, und wählen Sie dann Erstellen.
7. Gewähren Sie Zugriff auf Ihr Portfolio.
- a. Wählen Sie Ihr Portfolio auf der Portfolio-Seite in der AWS Service Catalog - Managementkonsole aus.
 - b. Erweitern Sie den Tab Gruppen, Rollen und Benutzer und wählen Sie Gruppen, Rollen, Benutzer hinzufügen aus.
 - c. Suchen Sie auf der Registerkarte Rollen nach Ihrer EMR-Studio-IAM-Rolle, wählen Sie Ihre Rolle aus und klicken Sie auf Zugriff hinzufügen.

Wenn Sie ...	Zugriff gewähren auf ...
IAM-Authentifizierung	Ihre nativen Benutzer
den IAM-Verbund	Ihre IAM-Rolle für den Verbund
den IAM-Identity-Center-Verbund	Ihre EMR-Studio-Benutzerrolle

Zugriff und Berechtigungen für Git-basierte Repositorys einrichten

EMR Studio unterstützt die folgenden Git-basierten Services:

- [AWS CodeCommit](#)
- [GitHub](#)
- [Bitbucket](#)
- [GitLab](#)

Damit EMR-Studio-Benutzer ein Git-Repository mit einem Workspace verknüpfen können, richten Sie die folgenden Zugriffs- und Berechtigungsanforderungen ein. Sie können auch Git-basierte Repositories konfigurieren, die Sie in einem privaten Netzwerk hosten, indem Sie den Anweisungen unter [Ein privat gehostetes Git-Repository für EMR Studio konfigurieren](#) folgen.

Cluster-Internetzugang

Sowohl Amazon EMR-Cluster, die auf Amazon EC2 ausgeführt werden, als auch Amazon EMR auf EKS-Clustern, die an Studio Workspaces angeschlossen sind, müssen sich in einem privaten Subnetz befinden, das ein Network Address Translation (NAT) -Gateway verwendet, oder sie müssen über ein Virtual Private Gateway auf das Internet zugreifen können. Weitere Informationen finden Sie unter [Amazon VPC-Optionen beim Starten eines Clusters](#).

Die Sicherheitsgruppen, die Sie mit EMR Studio verwenden, müssen auch eine ausgehende Regel enthalten, die es Workspaces ermöglicht, Datenverkehr von einem angeschlossenen EMR-Cluster ins Internet weiterzuleiten. Weitere Informationen finden Sie unter [Definieren Sie Sicherheitsgruppen zur Steuerung des Netzwerkverkehrs in EMR Studio](#).

Important

Wenn die Netzwerkschnittstelle in einem öffentlichem Subnetz befindet, kann sie nicht über ein Internet-Gateway (IGW) mit dem Internet kommunizieren.

Berechtigungen für AWS Secrets Manager

Um EMR-Studio-Benutzern den Zugriff auf Git-Repositories mit in AWS Secrets Manager gespeicherten Geheimnissen zu ermöglichen, fügen Sie der [Servicerolle für EMR Studio](#) eine Berechtigungsrichtlinie hinzu, die den Vorgang `secretsmanager:GetSecretValue` ermöglicht.

Informationen zum Verknüpfen von Git-basierten Repositories mit Workspaces finden Sie unter [Git-basierte Repositories mit einem EMR Studio Workspace verknüpfen](#).

Ein privat gehostetes Git-Repository für EMR Studio konfigurieren

Verwenden Sie die folgenden Anweisungen, um privat gehostete Repositories für Amazon EMR Studio zu konfigurieren. Sie müssen eine Konfigurationsdatei mit Informationen zu Ihren DNS- und Git-Servern bereitstellen. EMR Studio verwendet diese Informationen, um Workspaces zu konfigurieren, die den Datenverkehr an Ihre selbstverwalteten Repositories weiterleiten können.

 Note

Wenn Sie `DnsServerIPv4` konfigurieren, verwendet EMR Studio Ihren DNS-Server, um sowohl Ihren `GitServerDnsName` als auch Ihren Amazon-EMR-Endpunkt aufzulösen, z. B. `elasticmapreduce.us-east-1.amazonaws.com`. Um einen Endpunkt für Amazon EMR einzurichten, stellen Sie über die VPC, die Sie mit Ihrem Studio verwenden, eine Verbindung zu Ihrem Endpunkt her. Dadurch wird sichergestellt, dass der Amazon-EMR-Endpunkt zu einer privaten IP aufgelöst wird. Weitere Informationen finden Sie unter [Herstellen einer Verbindung mit Amazon EMR über einen Schnittstellen-VPC-Endpunkt](#).

Voraussetzungen

Bevor Sie ein privat gehostetes Git-Repository für EMR Studio konfigurieren, benötigen Sie einen Amazon-S3-Speicherort, an dem EMR Studio die Workspaces und Notebook-Dateien im Studio sichern kann. Verwenden Sie denselben S3-Bucket, den Sie beim Erstellen eines Studios angegeben haben.

Wie Sie ein oder mehrere privat gehostete Git-Repositorys für EMR Studio zu konfigurieren

1. Erstellen Sie eine Konfigurationsdatei mithilfe der folgenden Vorlage. Geben Sie für jeden Git-Server, den Sie in Ihrer Konfiguration angeben möchten, die folgenden Werte an:

- **DnsServerIPv4**- Die IPv4 Adresse Ihres DNS-Servers. Wenn Sie Werte für sowohl `DnsServerIPv4` als auch für `GitServerIPv4List` angeben, hat der Wert für `DnsServerIPv4` Vorrang und EMR Studio verwendet `DnsServerIPv4`, um Ihr `GitServerDnsName` zu lösen.

 Note

Um privat gehostete Git-Repositorys verwenden zu können, muss Ihr DNS-Server eingehenden Zugriff von EMR Studio zulassen. Wir bitten Sie dringend, Ihren DNS-Server vor anderen, unbefugten Zugriffen zu schützen.

- **GitServerDnsName** – Der DNS-Name Ihres Git-Servers. Zum Beispiel `"git.example.com"`.
- **GitServerIPv4List**- Eine Liste von IPv4 Adressen, die zu deinen Git-Servern gehören.

```
[
  {
    "Type": "PrivatelyHostedGitConfig",
    "Value": [
      {
        "DnsServerIPv4": "<10.24.34.xxx>",
        "GitServerDnsName": "<enterprise.git.com>",
        "GitServerIPv4List": [
          "<xxx.xxx.xxx.xxx>",
          "<xxx.xxx.xxx.xxx>"
        ]
      },
      {
        "DnsServerIPv4": "<10.24.34.xxx>",
        "GitServerDnsName": "<git.example.com>",
        "GitServerIPv4List": [
          "<xxx.xxx.xxx.xxx>",
          "<xxx.xxx.xxx.xxx>"
        ]
      }
    ]
  }
]
```

2. Speichern Sie Ihre Konfigurationsdatei unter `configuration.json`.
3. Laden Sie die Konfigurationsdatei in Ihren standardmäßigen Amazon S3 S3-Speicherort in einem Ordner mit dem Namen `hochlife-cycle-configuration`. Wenn Ihr Standard-S3-Speicherort beispielsweise `s3://amzn-s3-demo-bucket/workspace` lautet, befindet sich Ihre Konfigurationsdatei in `s3://amzn-s3-demo-bucket/workspace/life-cycle-configuration/configuration.json`.

Important

Wir bitten Sie dringend, den Zugriff auf Ihren `life-cycle-configuration`-Ordner auf Studio-Administratoren und Ihre EMR-Studio-Servicerolle zu beschränken und `configuration.json` vor unbefugtem Zugriff zu schützen. Anweisungen finden Sie unter [Steuern des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#) oder [Bewährte Sicherheitsmethoden für Amazon S3](#).

Anweisungen zum Hochladen finden Sie unter [Erstellen eines Ordners](#) und [Hochladen von Objekten](#) im Benutzerhandbuch für Amazon Simple Storage Service. Um Ihre Konfiguration auf einen vorhandenen Workspace anzuwenden, schließen Sie den Workspace und starten Sie ihn neu, nachdem Sie Ihre Konfigurationsdatei auf Amazon S3 hochgeladen haben.

Spark-Aufträge in EMR Studio optimieren

Wenn Sie einen Spark-Job mit EMR Studio ausführen, können Sie einige Schritte unternehmen, um sicherzustellen, dass Sie Ihre Amazon-EMR-Clusterressourcen optimieren.

Ihre Livy-Sitzung verlängern

Wenn Sie Apache Livy zusammen mit Spark auf Ihrem Amazon-EMR-Cluster verwenden, empfehlen wir Ihnen, Ihr Livy-Sitzungs-Timeout zu erhöhen, indem Sie einen der folgenden Schritte ausführen:

- Wenn Sie einen Amazon-EMR-Cluster erstellen, legen Sie diese Konfigurationsklassifizierung im Feld Konfiguration eingeben fest.

```
[
  {
    "Classification": "livy-conf",
    "Properties": {
      "livy.server.session.timeout": "8h"
    }
  }
]
```

- Stellen Sie für einen bereits laufenden EMR-Cluster eine Verbindung zu Ihrem Cluster mit ssh her und legen Sie die livy-conf Konfigurationsklassifizierung unter /etc/livy/conf/livy.conf fest.

```
[
  {
    "Classification": "livy-conf",
    "Properties": {
      "livy.server.session.timeout": "8h"
    }
  }
]
```

Möglicherweise müssen Sie Livy neu starten, nachdem Sie die Konfiguration geändert haben.

- Wenn Sie nicht möchten, dass es bei Ihrer Livy-Sitzung zu einem Timeout kommt, setzen Sie die Eigenschaft `livy.server.session.timeout-check` auf `false` in `/etc/livy/conf/livy.conf`.

Spark im Cluster-Modus ausführen

Im Clustermodus wird der Spark-Treiber auf einem Core-Knoten statt auf dem Primärknoten ausgeführt, wodurch die Ressourcennutzung auf dem Primärknoten verbessert wird.

Um Ihre Spark-Anwendung im Cluster-Modus statt im Standard-Client-Modus auszuführen, wählen Sie Cluster-Modus, wenn Sie bei der Konfiguration Ihres Spark-Schritts in Ihrem neuen Amazon-EMR-Cluster den Bereitstellungsmodus festlegen. Weitere Informationen finden Sie unter [Übersicht über den Clustermodus](#) in der Apache-Spark-Dokumentation.

Den Spark-Treiberspeicher erhöhen

Um den Speicher des Spark-Treibers zu erhöhen, konfigurieren Sie Ihre Spark-Sitzung mit dem `%configure` magischen Befehl in Ihrem EMR-Notebook, wie im folgenden Beispiel.

```
%configure -f  
{ "driverMemory": "6000M" }
```

Ein Amazon EMR Studio verwenden

Dieser Abschnitt enthält Themen, die Ihnen bei der Konfiguration und Interaktion mit einem Amazon EMR Studio helfen.

Das folgende Video enthält praktische Informationen wie das Erstellen eines neuen Workspace und das Starten eines neuen Amazon-EMR-Clusters mit einer Cluster-Vorlage. Das Video zeigt auch ein Beispiel-Notebook.

In diesem Abschnitt finden Sie die folgenden Themen, die Ihnen beim Arbeiten in EMR Studio helfen:

- [Lernen Sie EMR Studio-Arbeitsbereiche kennen](#)
- [Workspace-Zusammenarbeit in EMR Studio konfigurieren](#)
- [Einen EMR-Studio-Workspace mit einer Laufzeit-Rolle ausführen](#)

- [Führen Sie Amazon EMR Studio Workspace Workspace-Notebooks programmgesteuert aus](#)
- [Durchsuchen Sie Daten mit SQL Explorer für EMR Studio](#)
- [Einen Computer an einen EMR Studio Workspace anhängen](#)
- [Git-basierte Repositories mit einem EMR Studio Workspace verknüpfen](#)
- [Verwenden des Editors von Amazon Athena SQL in EMR Studio](#)
- [CodeWhisperer Amazon-Integration mit EMR Studio Workspaces](#)
- [Debuggen von Anwendungen und Aufträgen mit EMR Studio](#)
- [Kernel und Bibliotheken in einem EMR Studio Workspace installieren](#)
- [Verbessern Sie Kernel mit magic Befehle in EMR Studio](#)
- [Verwenden Sie mehrsprachige Notebooks mit Spark-Kernen](#)

Lernen Sie EMR Studio-Arbeitsbereiche kennen

Wenn Sie ein EMR Studio verwenden, können Sie verschiedene Workspaces erstellen und konfigurieren, um Notebooks zu organisieren und auszuführen. In diesem Abschnitt wird das Erstellen von und das Arbeiten mit Workspaces behandelt. Eine konzeptionelle Übersicht über die [Workspaces](#) finden Sie unter [Wie Amazon EMR Studio funktioniert](#).

In diesem Abschnitt finden Sie die folgenden Themen, die Ihnen beim Verwenden von EMR Studio Workspaces helfen:

- [Einen EMR Studio Workspace erstellen](#)
- [Starten Sie einen Workspace in EMR Studio](#)
- [Verstehen Sie die Workspace-Benutzeroberfläche in EMR Studio](#)
- [Entdecken Sie Notebook-Beispiele in einem EMR Studio-Workspace](#)
- [Workspace-Inhalte in EMR Studio speichern](#)
- [Löschen Sie einen Workspace und Notebook-Dateien in EMR Studio](#)
- [Informationen über Workspace-Status](#)
- [Beheben von Workspace-Verbindungsproblemen](#)

Einen EMR Studio Workspace erstellen

Sie können EMR Studio Workspaces erstellen, um Notebookcode mithilfe der-EMR Studio-Oberfläche auszuführen.

So erstellen Sie einen Workspace in einem EMR Studio

1. Melden Sie sich bei EMR Studio an.
2. Wählen Sie Workspace erstellen.
3. Geben Sie Workspace-Name und Beschreibung ein. Wenn Sie einen Workspace benennen, können Sie ihn auf der Seite Workspaces leichter identifizieren.
4. Wenn Sie in Echtzeit mit anderen Studio-Benutzern in diesem Workspace zusammenarbeiten möchten, aktivieren Sie die Workspace-Zusammenarbeit. Sie können Mitarbeiter konfigurieren, nachdem Sie den Workspace gestartet haben.
5. Wenn Sie einen Cluster an einen Workspace anhängen möchten, erweitern Sie den Abschnitt Erweiterte Konfiguration. Wenn Sie möchten, können Sie später einen Cluster anhängen. Weitere Informationen finden Sie unter [Einen Computer an einen EMR Studio Workspace anhängen](#).

Note

Um einen neuen Cluster bereitzustellen, benötigen Sie Zugriffsberechtigungen von Ihrem Administrator.

Wählen Sie eine der Clusteroptionen für den Workspace und hängen Sie den Cluster an. Weitere Informationen zum Bereitstellen eines Clusters beim Erstellen eines Workspace finden Sie unter [Einen neuen EMR-Cluster erstellen und an einen EMR Studio Workspace anhängen](#).

6. Wählen Sie unten rechts auf der Seite die Option Workspace erstellen aus.

Nachdem Sie einen Workspace erstellt haben, öffnet EMR Studio die Workspaces-Seite. Oben auf der Seite wird ein grünes Erfolgsbanner angezeigt und Sie können den neu erstellten Workspace in der Liste finden.

Standardmäßig wird ein Workspace geteilt und kann von allen Studio-Benutzern gesehen werden. Es kann jedoch jeweils nur ein Benutzer einen Workspace öffnen und darin arbeiten. Um gleichzeitig mit anderen Benutzern zu arbeiten, können Sie [Workspace-Zusammenarbeit in EMR Studio konfigurieren](#)

Starten Sie einen Workspace in EMR Studio

Um mit der Arbeit mit Notebookdateien zu beginnen, starten Sie einen Workspace, um auf den Notebook-Editor zuzugreifen. Auf der Seite Workspaces in einem Studio werden alle Workspaces aufgeführt, auf die Sie Zugriff haben, mit Details wie Name, Status, Erstellungszeit und Letzte Änderung.

Note

Wenn Sie EMR-Notebooks in der alten Amazon EMR-Konsole hatten, finden Sie sie in der Konsole als EMR Studio Workspaces. EMR Notebooks-Benutzer benötigen zusätzliche IAM-Rollenberechtigungen, um auf Workspaces zuzugreifen oder diese zu erstellen. Wenn Sie kürzlich ein Notizbuch in der alten Konsole erstellt haben, müssen Sie möglicherweise die Liste der Arbeitsbereiche aktualisieren, damit es in der Konsole angezeigt wird. Weitere Informationen zum Übergang finden Sie unter [Amazon EMR Notebooks sind als Amazon EMR Studio Workspaces in der Konsole verfügbar](#) und [Verwaltung von Amazon EMR-Clustern mit der Konsole](#)

So starten Sie einen Workspace zum Bearbeiten und Ausführen von Notebooks

1. Suchen Sie auf der Workspaces-Seite Ihres Studios nach dem Workspace. Sie können die Liste nach Schlüsselwort oder Spaltenwert filtern.
2. Wählen Sie den Workspace-Namen, um den Workspace in einer neuen Browser-Registerkarte zu starten. Es kann einige Minuten dauern, bis der Workspace geöffnet wird, wenn er inaktiv ist. Wählen Sie alternativ die Zeile für den Workspace aus und wählen Sie dann Workspace starten aus. Sie können aus den folgenden Startoptionen auswählen:
 - Schnellstart – Starten Sie Ihren Workspace schnell mit den Standardoptionen. Wählen Sie Schnellstart, wenn Sie Cluster an den Workspace in JupyterLab anhängen möchten.
 - Mit Optionen starten – Starten Sie Ihren Workspace mit benutzerdefinierten Optionen. Sie können entweder in Jupyter starten oder JupyterLab Ihren Workspace mit einem EMR-Cluster verbinden und Ihre Sicherheitsgruppen auswählen.

 Note

Es kann jeweils nur ein Benutzer einen Workspace öffnen und darin arbeiten. Wenn Sie einen Workspace auswählen, der bereits verwendet wird, zeigt EMR Studio eine Benachrichtigung an, wenn Sie versuchen, ihn zu öffnen. In der Spalte Benutzer auf der Workspaces-Seite wird der Benutzer angezeigt, der im Workspace arbeitet.

Verstehen Sie die Workspace-Benutzeroberfläche in EMR Studio

Die Benutzeroberfläche von EMR Studio Workspace basiert auf der [JupyterLabOberfläche](#) mit den mit Symbolen versehenen Tabs in der linken Seitenleiste. Wenn Sie den Mauszeiger über einem Symbol halten, wird ein Tooltip mit dem Namen der Registerkarte angezeigt. Wählen Sie in der linken Seitenleiste Registerkarten aus, um auf die folgenden Bereiche zuzugreifen.

- Dateibrowser – Zeigt die Dateien und Verzeichnisse im Workspace sowie die Dateien und Verzeichnisse der verknüpften Git-Repositorys an.
- Ausführen von Kernels und Terminals – Listet alle Kernel und Terminals auf, die im Workspace laufen. Weitere Informationen finden Sie in der offiziellen Dokumentation unter [Kernel und Terminals verwalten](#). JupyterLab
- Git – Stellt eine grafische Benutzeroberfläche für die Ausführung von Befehlen in den Git-Repositorys bereit, die an den Workspace angehängt sind. Dieses Panel ist eine JupyterLab Erweiterung namens jupyterlab-git. Weitere Informationen finden Sie unter [jupyterlab-git](#).
- EMR-Cluster – Ermöglicht das Anhängen eines Clusters an den Workspace oder das Trennen eines Clusters vom Workspace, um Notebookcode auszuführen. Das EMR-Cluster-Konfigurationsfenster bietet auch erweiterte Konfigurationsoptionen, mit denen Sie einen neuen Cluster erstellen und an den Workspace anhängen können. Weitere Informationen finden Sie unter [Einen neuen EMR-Cluster erstellen und an einen EMR Studio Workspace anhängen](#).
- Amazon EMR Git Repository – Hilft Ihnen, den Workspace mit bis zu drei Git-Repositorys zu verknüpfen. Weitere Informationen und Anweisungen finden Sie in [Git-basierte Repositorys mit einem EMR Studio Workspace verknüpfen](#).
- Notebook-Beispiele – Enthält eine Liste mit Notebookbeispielen, die Sie im Workspace speichern können. Sie können auf die Beispiele auch zugreifen, indem Sie auf der Launcher-Seite des Workspace die Option Notebook-Beispiele auswählen.

- **Befehle** — Bietet eine tastaturgesteuerte Möglichkeit, nach Befehlen zu suchen und diese auszuführen. JupyterLab Weitere Informationen finden Sie auf der Seite mit der [Befehlspalette](#) in der JupyterLab Dokumentation.
- **Notebook-Tools** – Ermöglicht die Auswahl und Einstellung von Optionen wie Zellfolientyp und Metadaten. Die Option Notebook-Tools wird in der linken Seitenleiste angezeigt, nachdem Sie eine Notebookdatei geöffnet haben.
- **Registerkarten öffnen** – Listet die geöffneten Dokumente und Aktivitäten im Haupt-Workspace auf, sodass Sie zu einer geöffneten Registerkarte springen können. Weitere Informationen finden Sie in der JupyterLab Dokumentation auf der Seite „[Tabs und Einzeldokumentmodus](#)“.
- **Zusammenarbeit** – Ermöglicht es Ihnen, die Zusammenarbeit im Workspace zu aktivieren oder zu deaktivieren und Mitarbeiter zu verwalten. Sie benötigen die nötigen Berechtigungen, um das Panel Zusammenarbeit anzeigen zu können. Weitere Informationen finden Sie unter [Legen Sie die Eigentümerschaft für die Workspace-Zusammenarbeit](#) fest.

Entdecken Sie Notebook-Beispiele in einem EMR Studio-Workspace

Jeder EMR Studio Workspace enthält eine Reihe von Notebookbeispielen, mit denen Sie die Features von EMR Studio erkunden können. Um ein Notebook-Beispiel zu bearbeiten oder auszuführen, können Sie es im Workspace speichern.

Um ein Notebookbeispiel in einem Workspace zu speichern

1. Wählen Sie in der linken Seitenleiste den Tab Notebook-Beispiele, um den Bereich Notebook-Beispiele zu öffnen. Sie können auf die Beispiele auch zugreifen, indem Sie auf der Launcher-Seite des Workspace die Option Notebook-Beispiele auswählen.
2. Wählen Sie ein Notebookbeispiel aus, um es im Hauptarbeitsbereich als Vorschau anzuzeigen. Das Beispiel ist schreibgeschützt.
3. Um das Notebookbeispiel im Workspace zu speichern, wählen Sie In Workspace speichern. EMR Studio speichert das Beispiel in Ihrem Stammverzeichnis. Nachdem Sie ein Notebook-Beispiel im Workspace gespeichert haben, können Sie es umbenennen, bearbeiten und ausführen.

Weitere Informationen zu den Notebook-Beispielen finden Sie im [EMR Studio Notebook Examples GitHub Repository](#).

Workspace-Inhalte in EMR Studio speichern

Wenn Sie im Notebook-Editor eines Workspace arbeiten, speichert EMR Studio den Inhalt der Notebookzellen und gibt ihn für Sie an dem Amazon-S3-Speicherort aus, der mit dem Studio verknüpft ist. Bei diesem Backup-Vorgang bleibt die Arbeit zwischen den Sitzungen erhalten.

Sie können ein Notebook auch speichern, indem Sie auf der Registerkarte Notebook öffnen STRG+S drücken oder eine der Speicheroptionen unter Datei verwenden.

Eine weitere Möglichkeit, die Notebookdateien in einem Workspace zu sichern, besteht darin, den Workspace mit einem Git-basierten Repository zu verknüpfen und Ihre Änderungen mit dem Remote-Repository zu synchronisieren. Auf diese Weise können Sie auch Notizbücher speichern und mit Teammitgliedern teilen, die einen anderen Workspace oder Studio verwenden. Detaillierte Anweisungen finden Sie unter [Git-basierte Repositories mit einem EMR Studio Workspace verknüpfen](#).

Löschen Sie einen Workspace und Notebook-Dateien in EMR Studio

Wenn Sie eine Notebookdatei aus einem EMR Studio-Arbeitsbereich löschen, löschen Sie die Datei aus dem Dateibrowser, und EMR Studio entfernt ihre Sicherungskopie in Amazon S3. Sie müssen keine weiteren Schritte unternehmen, um Speichergebühren zu vermeiden, wenn Sie eine Datei aus einem Workspace löschen.

Wenn Sie einen gesamten Workspace löschen, verbleiben die zugehörigen Notebookdateien und -ordner am Amazon-S3-Speicherort. Für die Dateien fallen weiterhin Speichergebühren an. Um Speichergebühren zu vermeiden, entfernen Sie alle gesicherten Dateien und Ordner, die mit Ihrem gelöschten Workspace verknüpft sind, aus Amazon S3.

So löschen Sie eine Notebook-Datei aus einem EMR Studio Workspace

1. Wählen Sie in der linken Seitenleiste des Workspace den Bereich Dateibrowser aus.
2. Wählen Sie die Datei oder den Ordner aus, die Sie löschen möchten. Klicken Sie mit der rechten Maustaste auf die ausgewählten Dateien oder Ordner und wählen Sie Löschen. Die Datei verschwindet aus der Liste. EMR Studio entfernt die Datei oder den Ordner für Sie aus Amazon S3.

From the Workspace UI

Löschen Sie einen Workspace und die zugehörigen Backup-Dateien aus EMR Studio

1. Melden Sie sich mit Ihrer Studio-Zugriffs-URL bei Ihrem EMR Studio an und wählen Sie in der linken Navigationsleiste Workspaces aus.
2. Suchen Sie in der Liste nach Ihrem Workspace und aktivieren Sie das Kontrollkästchen neben dessen Namen. Sie können mehrere Arbeitsbereiche zum gleichzeitigen Löschen auswählen.
3. Wählen Sie oben rechts in der Liste der Arbeitsbereiche die Option Löschen aus und bestätigen Sie, dass Sie die ausgewählten Arbeitsbereiche löschen möchten. Wählen Sie zur Bestätigung Delete.
4. Wenn Sie die Notebookdateien, die mit dem gelöschten Workspace verknüpft waren, aus Amazon S3 entfernen möchten, folgen Sie den Anweisungen zum [Löschen von Objekten](#) im Konsole-Benutzerhandbuch von Amazon Simple Storage Service. Wenn Sie das Studio nicht erstellt haben, wenden Sie sich an Ihren Studio-Administrator, um den Amazon-S3-Backup-Speicherort für den gelöschten Workspace zu ermitteln.

From the Workspaces list

Löschen Sie einen Workspace und die zugehörigen Backup-Dateien aus der Workspaces-Liste

1. Navigieren Sie in der Konsole zur Workspace-Liste.
2. Wählen Sie den Workspace, den Sie löschen möchten, aus der Liste aus und klicken Sie dann auf Aktionen.
3. Wählen Sie Löschen.
4. Wenn Sie die Notebookdateien, die mit dem gelöschten Workspace verknüpft waren, aus Amazon S3 entfernen möchten, folgen Sie den Anweisungen zum [Löschen von Objekten](#) im Konsole-Benutzerhandbuch von Amazon Simple Storage Service. Wenn Sie das Studio nicht erstellt haben, wenden Sie sich an Ihren Studio-Administrator, um den Amazon-S3-Backup-Speicherort für den gelöschten Workspace zu ermitteln.

Informationen über Workspace-Status

Nachdem Sie einen EMR-Studio-Workspace erstellt haben, wird er als Zeile in der Workspaces-Liste in Ihrem Studio mit seinem Namen, Status, Erstellungszeit und dem Zeitstempel der letzten Änderung angezeigt. Die folgende Tabelle beschreibt den Workspace-Status.

Status	Description
Wird gestartet	Der Workspace wird vorbereitet, ist aber noch nicht einsatzbereit. Sie können einen Workspace nicht öffnen, wenn er den Status „Wird gestartet“ hat.
Bereit	Sie können den Workspace öffnen, um den Notebook-Editor zu verwenden, aber Sie müssen den Workspace an einen EMR-Cluster anhängen, bevor Sie Notebookcode ausführen können.
Anfügen	Der Workspace wird an einen Cluster angehängt.
Attached (Angefügt)	Der Workspace ist an einen EMR-Cluster angeschlossen und bereit, damit Sie Notebookcode schreiben und ausführen können. Wenn der Status eines Workspaces nicht Angefügt lautet, müssen Sie ihn an einen Cluster anhängen, bevor Sie Notebook-Code ausführen können.
Inaktiv	Der Workspace wurde gestoppt. Um einen inaktiven Workspace zu reaktivieren, wählen Sie ihn aus der Liste Workspaces aus. Wenn Sie den Workspace auswählen, ändert sich der Status von Inaktiv zu Bereit.
Wird angehalten	Der Workspace wird heruntergefahren und auf Inaktiv gesetzt. Wenn Sie einen

Status	Description
	Workspace beenden, beendet er alle entsprechenden Notebook-Kernel. EMR Studio stoppt Notebooks, die lange Zeit inaktiv waren.
Wird gelöscht	Wenn Sie einen Workspace löschen, markiert EMR Studio ihn zum Löschen und startet den Löschvorgang. Nach Abschluss des Löschvorgangs verschwindet der Workspace aus der Liste. Wenn Sie einen Workspace löschen, verbleiben seine Notebookdateien im Amazon-S3-Speicherort.

Beheben von Workspace-Verbindungsproblemen

Um Probleme mit der Workspace-Konnektivität zu lösen, können Sie einen Workspace beenden und neu starten. Wenn Sie einen Workspace neu starten, startet EMR Studio den Workspace in einer anderen Availability Zone oder einem anderen Subnetz, das mit Ihrem Studio verknüpft ist.

So beenden Sie einen EMR Studio Workspace und starten ihn neu

1. Schließen Sie den Workspace in Ihrem Browser.
2. Navigieren Sie in der Konsole zur Workspace-Liste.
3. Wählen Sie Ihren Workspace aus der Liste aus und klicken Sie auf Aktionen.
4. Wählen Sie Stopp und warten Sie, bis sich der Workspace-Status von Stopp auf Inaktiv ändert.
5. Wählen Sie erneut Aktionen und dann Start, um den Workspace neu zu starten.
6. Warten Sie, bis sich der Workspace-Status von Beginnt auf Bereit ändert, und wählen Sie dann den Workspace-Namen, um ihn in einer neuen Browser-Registerkarte erneut zu öffnen.

Workspace-Zusammenarbeit in EMR Studio konfigurieren

Mit Workspace-Zusammenarbeit können Sie Notebook-Code gleichzeitig mit anderen Mitgliedern Ihres Teams schreiben und ausführen. Wenn Sie an derselben Notebookdatei arbeiten, sehen Sie die Änderungen, die Ihre Mitarbeiter vornehmen. Sie können die Zusammenarbeit aktivieren, wenn

Sie einen Workspace erstellen, oder die Zusammenarbeit in einem vorhandenen Workspace ein- und ausschalten.

 Note

Die Zusammenarbeit mit EMR Studio Workspace wird bei [interaktiven EMR-Serverless-Anwendungen](#), oder wenn die Verbreitung vertrauenswürdiger Identitäten aktiviert ist, nicht unterstützt.

Voraussetzungen

Bevor Sie die Zusammenarbeit für einen Workspace konfigurieren, stellen Sie sicher, dass Sie die folgenden Aufgaben abgeschlossen haben:

- Stellen Sie sicher, dass Ihr EMR-Studio-Administrator Ihnen die erforderlichen Berechtigungen erteilt hat. Die folgende Beispielanweisung ermöglicht es einem Benutzer, die Zusammenarbeit für jeden Workspace mit dem Tag-Schlüssel `creatorUserId` zu konfigurieren, dessen Wert der Benutzer-ID entspricht (angegeben durch die RichtlinienvARIABLE `aws:userId`).

```
{
  "Sid": "UserRolePermissionsForCollaboration",
  "Action": [
    "elasticmapreduce:UpdateEditor",
    "elasticmapreduce:PutWorkspaceAccess",
    "elasticmapreduce>DeleteWorkspaceAccess",
    "elasticmapreduce:ListWorkspaceAccessIdentities"
  ],
  "Resource": "*",
  "Effect": "Allow",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
    }
  }
}
```

- Stellen Sie sicher, dass die mit Ihrem EMR Studio verknüpfte Servicerolle über die erforderlichen Berechtigungen verfügt, um die Workspace-Zusammenarbeit zu aktivieren und zu konfigurieren, wie in der folgenden Beispielanweisung dargestellt.

```
{
  "Sid": "AllowWorkspaceCollaboration",
  "Effect": "Allow",
  "Action": [
    "iam:GetUser",
    "iam:GetRole",
    "iam:ListUsers",
    "iam:ListRoles",
    "sso:GetManagedApplicationInstance",
    "sso-directory:SearchUsers"
  ],
  "Resource": "*"
}
```

Weitere Informationen finden Sie unter [Eine EMR-Studio-Servicerolle erstellen](#).

Um die Workspace-Zusammenarbeit zu aktivieren und Mitarbeiter hinzuzufügen

1. Wähle in deinem Workspace im Launcher-Bildschirm oder unten im linken Bereich das Zusammenarbeit-Symbol aus.

 Note

Das Panel Zusammenarbeit wird Ihnen nur angezeigt, wenn Ihr Studio-Administrator Ihnen die Erlaubnis erteilt hat, die Zusammenarbeit für den Workspace zu konfigurieren. Weitere Informationen finden Sie unter [Legen Sie die Eigentümerschaft für die Workspace-Zusammenarbeit](#) fest.

2. Vergewissern Sie sich, dass der Schalter Workspace-Zusammenarbeit zulassen aktiviert ist. Wenn Sie die Zusammenarbeit aktivieren, können nur Sie und die von Ihnen hinzugefügten Mitarbeiter den Workspace in der Liste auf der Studio-Workspaces-Seite sehen.
3. Geben Sie einen Namen für den Mitarbeiter ein. In Ihrem Workspace können maximal fünf Mitarbeiter enthalten sein, einschließlich Ihnen. Ein Mitarbeiter kann jeder Benutzer sein, der Zugriff auf Ihr EMR Studio hat. Wenn Sie keinen Mitarbeiter angeben, ist der Workspace ein privater Workspace, auf den nur Sie zugreifen können.

In der folgenden Tabelle sind die zutreffenden Werte für Mitarbeiter angegeben, die je nach Identitätstyp des Inhabers eingegeben werden müssen.



Note

Ein Inhaber kann nur Mitarbeiter mit demselben Identitätstyp einladen. Beispielsweise kann ein Benutzer nur andere Benutzer hinzufügen, und ein IAM-Identity-Center-Benutzer kann nur andere IAM-Identity-Center-Benutzer hinzufügen.

Authentifizierungsmodus	Wert, der für den Namen des Mitarbeiters eingegeben werden soll
IAM-Authentifizierung	Ein Benutzername. Dies ist der Name, den ein Benutzer sieht, wenn er bei der AWS Management Console angemeldet ist.
den IAM-Verbund	Der Name einer IAM-Rolle und ein optionaler Sitzungsname. Um alle Verbundbenutzer hinzuzufügen, die dieselbe IAM-Rolle annehmen, geben Sie den Namen einer IAM-Rolle für den Verbund an. Um einen einzelnen Benutzer als Mitarbeit er hinzuzufügen, geben Sie eine Rolle und einen Sitzungsnamen an. Beispiel, <code>MyRoleName:MySessionName</code> .
SSO	Ein IAM-Identity-Center-Benutzername wie <code>user@example.com</code> .

4. Wählen Sie Hinzufügen aus. Der Mitarbeiter kann den Workspace jetzt auf seiner EMR-Studio-Workspaces-Seite sehen und den Workspace starten, um ihn in Echtzeit mit Ihnen zu verwenden.

 Note

Wenn Sie die Workspace-Zusammenarbeit deaktivieren, kehrt der Workspace in seinen gemeinsamen Status zurück und kann von allen Studio-Benutzern eingesehen werden. Im geteilten Status kann jeweils nur ein Studio-Benutzer den Workspace öffnen und darin arbeiten.

Einen EMR-Studio-Workspace mit einer Laufzeit-Rolle ausführen

 Note

Die auf dieser Seite beschriebene Runtime-Rollenfunktionalität gilt nur für Amazon EMR, das auf Amazon ausgeführt wird EC2, und bezieht sich nicht auf die Runtime-Rollenfunktionalität in serverlosen interaktiven EMR-Anwendungen. Weitere Informationen zur Verwendung von Laufzeit-Rollen in EMR Serverless finden Sie unter [Auftrags-Laufzeit-Rollen](#) im Benutzerhandbuch zu Amazon EMR Serverless.

Eine Runtime-Rolle ist eine AWS Identity and Access Management (IAM) -Rolle, die Sie angeben können, wenn Sie einen Job oder eine Abfrage an einen Amazon EMR-Cluster senden. Der Job oder die Abfrage, die Sie an Ihren EMR-Cluster senden, verwendet die Runtime-Rolle, um auf AWS Ressourcen wie Objekte in Amazon S3 zuzugreifen.

Wenn Sie einen EMR Studio-Workspace an einen EMR-Cluster anhängen, der Amazon EMR 6.11 oder höher verwendet, können Sie eine Runtime-Rolle für den Job oder die Abfrage auswählen, die Sie einreichen, um sie beim Zugriff auf Ressourcen zu verwenden. AWS Wenn der EMR-Cluster jedoch keine Runtime-Rollen unterstützt, übernimmt der EMR-Cluster die Rolle nicht, wenn er auf Ressourcen zugreift AWS .

Bevor Sie eine Laufzeit-Rolle mit einem Amazon-EMR-Studio-Workspace verwenden können, muss ein Administrator Benutzerberechtigungen konfigurieren, sodass der Studio-Benutzer die `elasticmapreduce:GetClusterSessionCredentials`-API für die Laufzeit-Rolle aufrufen kann. Starten Sie dann einen neuen Cluster mit einer Laufzeit-Rolle, die Sie mit Ihrem Amazon EMR Studio Workspace verwenden können.

Auf dieser Seite

- [Konfigurieren Sie Benutzerberechtigungen für die Laufzeit-Rolle](#)

- [Starten Sie einen neuen Cluster mit einer Laufzeit-Rolle](#)
- [Verwenden Sie den EMR-Cluster mit einer Laufzeit-Rolle in Workspaces](#)
- [Überlegungen](#)

Konfigurieren Sie Benutzerberechtigungen für die Laufzeit-Rolle

Konfigurieren Sie Benutzerberechtigungen, sodass der Studio-Benutzer die `elasticmapreduce:GetClusterSessionCredentials`-API für die Laufzeit-Rolle aufrufen kann, die der Benutzer verwenden möchte. Sie müssen auch [the section called “Studio-Benutzerberechtigungen \(EC2, EKS\)”](#) konfigurieren, bevor der Benutzer Studio verwenden kann.

Warning

Um diese Berechtigung zu erteilen, erstellen Sie eine auf dem `elasticmapreduce:ExecutionRoleArn` Kontextschlüssel basierende Bedingung, wenn Sie einem Anrufer Zugriff auf den gewähren. `GetClusterSessionCredentials` APIs Das folgende Beispiel veranschaulicht die Vorgehensweise hierfür.

```
{
  "Sid": "AllowSpecificExecRoleArn",
  "Effect": "Allow",
  "Action": [
    "elasticmapreduce:GetClusterSessionCredentials"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ExecutionRoleArn": [
        "arn:aws:iam::111122223333:role/test-emr-demo1",
        "arn:aws:iam::111122223333:role/test-emr-demo2"
      ]
    }
  }
}
```

Das folgende Beispiel zeigt, wie einem IAM-Prinzipal die Verwendung einer IAM-Rolle mit dem Namen `test-emr-demo3`-Laufzeit-Rolle ermöglicht wird. Darüber hinaus kann der Versicherungsnehmer nur mit der Cluster-ID `j-123456789` auf Amazon-EMR-Cluster zugreifen.

```
{
  "Sid": "AllowSpecificExecRoleArn",
  "Effect": "Allow",
  "Action": [
    "elasticmapreduce:GetClusterSessionCredentials"
  ],
  "Resource": [
    "arn:aws:elasticmapreduce:<region>:111122223333:cluster/j-123456789"
  ],
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ExecutionRoleArn": [
        "arn:aws:iam::111122223333:role/test-emr-demo3"
      ]
    }
  }
}
```

Im folgenden Beispiel kann ein IAM-Prinzipal jede IAM-Rolle, deren Name mit der Zeichenfolge `test-emr-demo4` beginnt, als Laufzeitrolle verwenden. Darüber hinaus kann der Versicherungsnehmer nur auf Amazon-EMR-Cluster zugreifen, die mit dem Schlüssel-Wert-Paar `tagKey: tagValue` gekennzeichnet sind.

```
{
  "Sid": "AllowSpecificExecRoleArn",
  "Effect": "Allow",
  "Action": [
    "elasticmapreduce:GetClusterSessionCredentials"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ResourceTag/tagKey": "tagValue"
    },
    "StringLike": {
      "elasticmapreduce:ExecutionRoleArn": [
        "arn:aws:iam::111122223333:role/test-emr-demo4*"
      ]
    }
  }
}
```

Starten Sie einen neuen Cluster mit einer Laufzeit-Rolle

Nachdem Sie über die erforderlichen Berechtigungen verfügen, starten Sie einen neuen Cluster mit einer Laufzeit-Rolle, die Sie mit Ihrem Amazon EMR Studio Workspace verwenden können.

Wenn Sie bereits einen neuen Cluster mit einer Laufzeit-Rolle gestartet haben, können Sie mit dem Abschnitt [the section called “Verwenden Sie den Cluster mit Ihrem Workspace”](#) fortfahren.

1. Erfüllen Sie zunächst die Voraussetzungen im Abschnitt [Schritte für Laufzeit-Rollen für Amazon EMR](#).
2. Starten Sie dann einen Cluster mit den folgenden Einstellungen, um Laufzeit-Rollen mit Amazon EMR Studio Workspaces zu verwenden. Anweisungen zum Start Ihres Clusters finden Sie unter [Geben Sie eine Sicherheitskonfiguration für einen Amazon EMR-Cluster an](#).
 - Wählen Sie für Release die Option emr-6.11.0 oder höher aus.
 - Wählen Sie Spark, Livy und Jupyter Enterprise Gateway als Ihre Cluster-Anwendungen aus.
 - Verwenden Sie die Sicherheitskonfiguration, die Sie im vorherigen Schritt erstellt haben.
 - Optional können Sie Lake Formation für Ihren EMR-Cluster aktivieren. Weitere Informationen finden Sie unter [Wie Amazon EMR mit Lake Formation funktioniert](#).

Nachdem Sie Ihren Cluster gestartet haben, können Sie [den rollenfähigen Laufzeit-Cluster mit einem EMR Studio Workspace verwenden](#).

Note

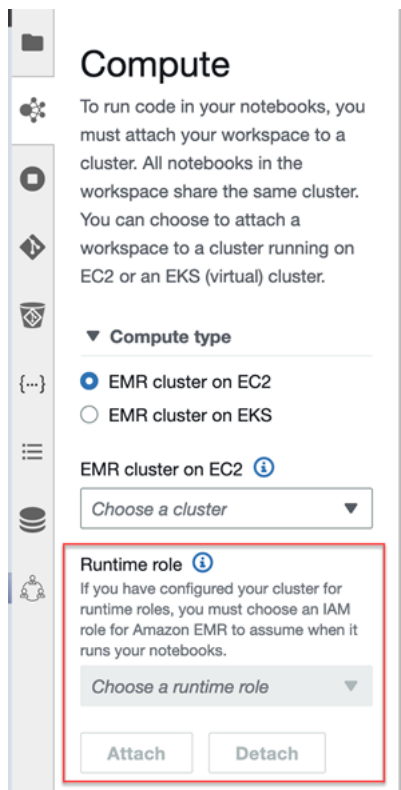
Der [ExecutionRoleArn](#)Wert wird derzeit bei der [StartNotebookExecution](#)API-Operation nicht unterstützt, obwohl der ExecutionEngineConfig.Type WertEMR.

Verwenden Sie den EMR-Cluster mit einer Laufzeit-Rolle in Workspaces

Sobald Sie Ihren Cluster eingerichtet und gestartet haben, können Sie den rollenfähigen Laufzeit-Cluster mit Ihrem EMR Studio Workspace verwenden.

1. Erstellen Sie einen neuen Workspace oder starten Sie einen vorhandenen Workspace. Weitere Informationen finden Sie unter [Einen EMR Studio Workspace erstellen](#).

2. Wählen Sie in der linken Seitenleiste Ihres geöffneten Workspace die Registerkarte EMR-Cluster, erweitern Sie den Abschnitt Compute-Typ und wählen Sie Ihren Cluster aus dem EMR-Cluster EC2 im Menü und die Runtime-Rolle aus dem Runtime-Rollenmenü aus.



3. Wählen Sie Anhängen, um den Cluster mit der Laufzeit-Rolle an Ihren Workspace anzuhängen.

Note

Beachten Sie bei der Auswahl einer Runtime-Rolle, dass ihr zugrunde liegende verwaltete Richtlinien zugeordnet sein können. In den meisten Fällen empfehlen wir, begrenzte Ressourcen zu wählen, z. B. bestimmte Notizbücher. Wenn Sie sich beispielsweise für eine Runtime-Rolle entscheiden, die den Zugriff auf all Ihre Notizbücher beinhaltet, gewährt die verwaltete Richtlinie, die der Rolle zugeordnet ist, vollen Zugriff.

Überlegungen

Beachten Sie die folgenden Überlegungen, wenn Sie einen rollenfähigen Laufzeit-Cluster mit Ihrem Amazon EMR Studio Workspace verwenden:

- Sie können eine Laufzeit-Rolle nur auswählen, wenn Sie einen EMR Studio Workspace an einen EMR-Cluster anhängen, der Amazon-EMR-Version 6.11 oder höher verwendet.
- Die auf dieser Seite beschriebene Runtime-Rollenfunktionalität wird nur unterstützt, wenn Amazon EMR auf Amazon ausgeführt wird EC2, und sie wird nicht von serverlosen interaktiven EMR-Anwendungen unterstützt. Weitere Informationen zu Laufzeit-Rollen für EMR Serverless finden Sie unter [Auftrags-Laufzeit-Rollen](#) im Benutzerhandbuch zu Amazon EMR Serverless.
- Sie müssen zwar zusätzliche Berechtigungen konfigurieren, bevor Sie eine Laufzeit-Rolle angeben können, wenn Sie einen Job an einen Cluster senden, aber Sie benötigen keine zusätzlichen Berechtigungen, um auf die von einem EMR Studio Workspace generierten Dateien zuzugreifen. Die Berechtigungen für solche Dateien sind dieselben wie für Dateien, die aus Clustern ohne Laufzeit-Rollen generiert wurden.
- Sie können SQL Explorer nicht in einem EMR Studio Workspace mit einem Cluster verwenden, der über eine Laufzeit-Rolle verfügt. Amazon EMR deaktiviert SQL Explorer in der Benutzeroberfläche, wenn ein Workspace an einen EMR-Cluster mit Laufzeit-Rolle angehängt ist.
- Sie können den Kollaborationsmodus nicht in einem EMR-Studio-Workspace mit einem Cluster verwenden, der über eine Laufzeit-Rolle verfügt. Amazon EMR deaktiviert die Funktionen für die Zusammenarbeit in Workspace, wenn ein Workspace an einen EMR-Cluster mit aktivierter Laufzeit-Rolle angehängt ist. Der Workspace bleibt nur für den Benutzer zugänglich, der den Workspace angehängt hat.
- Sie können Laufzeitrollen nicht in einem Studio verwenden, in dem die Verbreitung vertrauenswürdiger Identitäten von IAM Identity Center aktiviert ist.
- Möglicherweise wird die Warnung Die Seite ist möglicherweise nicht sicher! angezeigt von der Spark-Benutzeroberfläche für einen rollenfähigen Runtime-Cluster, der Amazon EMR Version 7.4.0 und niedriger verwendet. In diesem Fall umgehen Sie die Warnung, um weiterhin die Spark-Benutzeroberfläche zu sehen.

Führen Sie Amazon EMR Studio Workspace Workspace-Notebooks programmgesteuert aus

Note

Die programmatische Ausführung von Notebooks wird mit interaktiven Amazon-EMR-Serverless-Anwendungen nicht unterstützt.

Sie können Ihre Amazon EMR Studio Workspace-Notebooks programmgesteuert mit einem Skript oder auf AWS CLI ausführen. Informationen zum programmgesteuerten Ausführen Ihres Notebooks finden Sie unter [Beispiele für programmatische Befehle für EMR Notebooks](#).

Durchsuchen Sie Daten mit SQL Explorer für EMR Studio

Note

SQL Explorer für EMR Studio wird mit interaktiven Anwendungen von Amazon EMR Serverless oder in einem Studio, in dem die Verbreitung vertrauenswürdiger Identitäten von IAM Identity Center aktiviert ist, nicht unterstützt.

In diesem Thema finden Sie Informationen, die Ihnen bei den ersten Schritten mit dem SQL Explorer in Amazon EMR Studio helfen. SQL Explorer ist ein einseitiges Tool in Ihrem Workspace, das Ihnen hilft, die Datenquellen im Datenkatalog Ihres EMR-Clusters zu verstehen. Sie können SQL Explorer verwenden, um Ihre Daten zu durchsuchen, SQL-Abfragen zum Abrufen von Daten auszuführen und Abfrageergebnisse herunterzuladen.

SQL Explorer unterstützt Presto. Bevor Sie SQL Explorer verwenden, stellen Sie sicher, dass Sie über einen Cluster verfügen, der Amazon-EMR-Version 5.34.0 oder höher oder Version 6.4.0 oder höher verwendet und Presto installiert hat. Der Amazon EMR Studio SQL Explorer unterstützt keine Presto-Cluster, die Sie mit Verschlüsselung während der Übertragung konfiguriert haben. Das liegt daran, dass Presto auf diesen Clustern im TLS-Modus ausgeführt wird.

Durchsuchen Sie den Datenkatalog Ihres Clusters

SQL Explorer bietet eine Katalogbrowser-Oberfläche, mit der Sie untersuchen und verstehen können, wie Ihre Daten organisiert sind. Sie können beispielsweise den Datenkatalogbrowser verwenden, um Tabellen- und Spaltennamen zu überprüfen, bevor Sie eine SQL-Abfrage schreiben.

Wie Sie Ihren Datenkatalog durchsuchen

1. Öffnen Sie SQL Explorer in Ihrem Workspace.
2. Stellen Sie sicher, dass Ihr Workspace mit einem EMR-Cluster verbunden ist EC2, auf dem Amazon EMR Version 6.4.0 oder höher mit installiertem Presto ausgeführt wird. Sie können einen vorhandenen Cluster auswählen oder einen neuen erstellen. Weitere Informationen finden Sie unter [Einen Computer an einen EMR Studio Workspace anhängen](#).
3. Wählen Sie eine Datenbank aus der Drop-down-Liste aus, um sie zu durchsuchen.

4. Erweitern Sie eine Tabelle in Ihrer Datenbank, um die Spaltennamen der Tabelle zu sehen. Sie können in der Suchleiste auch ein Schlüsselwort eingeben, um die Tabellenergebnisse zu filtern.

Führen Sie eine SQL-Abfrage aus, um Daten abzurufen

Um Daten mit einer SQL-Abfrage abzurufen und die Ergebnisse herunterzuladen

1. Öffnen Sie SQL Explorer in Ihrem Workspace.
2. Stellen Sie sicher, dass Ihr Workspace an einen EMR-Cluster angeschlossen ist, auf EC2 dem Presto und Spark installiert sind. Sie können einen vorhandenen Cluster auswählen oder einen neuen erstellen. Weitere Informationen finden Sie unter [Einen Computer an einen EMR Studio Workspace anhängen](#).
3. Wählen Sie Editor öffnen, um eine neue Editor-Registerkarte in Ihrem Workspace zu öffnen.
4. Verfassen Sie Ihre SQL-Abfrage auf der Registerkarte „Editor“.
5. Wählen Sie Ausführen aus.
6. Sehen Sie sich Ihre Abfrageergebnisse unter Ergebnisvorschau an. SQL Explorer zeigt standardmäßig die ersten 100 Ergebnisse an. Sie können eine andere Anzahl von Ergebnissen für die Anzeige auswählen (bis zu 1 000), indem Sie das Dropdownmenü Vorschau der ersten 100 Abfrageergebnisse verwenden.
7. Wählen Sie Ergebnisse herunterladen, um Ihre Ergebnisse im CSV-Format herunterzuladen. Sie können bis zu 1 000 Ergebniszeilen herunterladen.

Einen Computer an einen EMR Studio Workspace anhängen

Amazon EMR Studio führt Notebook-Befehle mithilfe eines Kernels auf einem EMR-Cluster aus. Bevor Sie einen Kernel auswählen können, sollten Sie den Workspace an einen Cluster anhängen, der EC2 Amazon-Instances verwendet, an einen Amazon EMR-on-EKS-Cluster oder an eine EMR-Serverless-Anwendung. Mit EMR Studio können Sie Workspaces an neue oder bestehende Cluster anhängen und haben die Flexibilität, Cluster zu ändern, ohne den Workspace schließen zu müssen.

In diesem Abschnitt finden Sie die folgenden Themen, die Ihnen beim Arbeiten mit und beim Bereitstellen von Clustern für EMR Studio helfen:

- [Einen EC2 Amazon-Cluster an einen EMR Studio Workspace anhängen](#)
- [Einen Amazon EMR in EKS-Cluster an einen EMR-Studio-Workspace anhängen](#)
- [Eine Amazon-EMR-Serverless-Anwendung an einen EMR Studio Workspace anhängen](#)

- [Einen neuen EMR-Cluster erstellen und an einen EMR Studio Workspace anhängen](#)
- [Trennen Sie einen Computer von einem EMR Studio Workspace](#)

Einen EC2 Amazon-Cluster an einen EMR Studio Workspace anhängen

Sie können einen auf Amazon ausgeführten EMR-Cluster EC2 an einen Workspace anhängen, wenn Sie den Workspace erstellen, oder einen Cluster an einen vorhandenen Workspace anhängen. Wenn Sie einen neuen Cluster erstellen und anhängen möchten, lesen Sie [Einen neuen EMR-Cluster erstellen und an einen EMR Studio Workspace anhängen](#).

Note

Ein Workspace in einem Studio, für den die Verbreitung vertrauenswürdiger Identitäten von IAM Identity Center aktiviert ist, kann nur eine Verbindung zu einem EMR-Cluster mit einer Sicherheitskonfiguration herstellen, für die Identity Center aktiviert ist.

On create

Beim Erstellen eines Workspace eine Verbindung zu einem Amazon-EMR-Compute-Cluster herstellen

1. Stellen Sie im Dialogfeld Workspace erstellen sicher, dass Sie bereits ein Subnetz für den neuen Workspace ausgewählt haben. Erweitern Sie den Abschnitt Erweiterte Konfiguration.
2. Wählen Sie Workspace an einen EMR-Cluster anhängen.
3. Wählen Sie in der EMR-Cluster-Dropdown-Liste einen vorhandenen EMR-Cluster aus, der an den Workspace angehängt werden soll.

Nachdem Sie einen Cluster angehängt haben, beenden Sie die Erstellung des Workspace. Wenn Sie den neuen Workspace zum ersten Mal öffnen und den Bereich EMR-Cluster auswählen, sollte Ihr ausgewählter Cluster angehängt sein.

On launch

Stellen Sie eine Verbindung zu einem Amazon-EMR-Rechencluster her, wenn Sie den Workspace starten

1. Navigieren Sie zur Workspaces-Liste und wählen Sie die Zeile für den Workspace aus, den Sie starten möchten. Wählen Sie dann Workspace starten > Mit Optionen starten aus.
2. Wählen Sie einen EMR-Cluster aus, der an Ihren Workspace angehängt werden soll.

Nachdem Sie einen Cluster angehängt haben, beenden Sie die Erstellung des Workspace. Wenn Sie den neuen Workspace zum ersten Mal öffnen und den Bereich EMR-Cluster auswählen, sollte Ihr ausgewählter Cluster angehängt sein.

In JupyterLab

Einen Workspace an einen Amazon EMR-Compute-Cluster anhängen in JupyterLab

1. Wählen Sie Ihren Workspace und dann Workspace starten > Schnellstart.
2. Öffnen Sie im Inneren JupyterLab die Registerkarte Cluster in der linken Seitenleiste.
3. Wählen Sie das Drop-down-Menü EMR on EC2 Cluster oder wählen Sie einen Amazon EMR on EKS-Cluster aus.
4. Wählen Sie Anfügen, um den Cluster an Ihren Workspace anzufügen.

Nachdem Sie einen Cluster angehängt haben, beenden Sie die Erstellung des Workspace. Wenn Sie den neuen Workspace zum ersten Mal öffnen und den Bereich EMR-Cluster auswählen, sollte Ihr ausgewählter Cluster angehängt sein.

In the Workspace UI

Hängen Sie über die Workspace-Benutzeroberfläche einen Workspace an einen Amazon-EMR-Compute-Cluster an

1. Wählen Sie in dem Workspace, den Sie einem Cluster zuordnen möchten, in der linken Seitenleiste das EMR-Cluster-Symbol aus, um das Cluster-Bereich zu öffnen.
2. Erweitern Sie unter Clustertyp die Dropdownliste und wählen Sie EMR-Cluster on aus. EC2
3. Wählen Sie Cluster in der Dropdown-Liste aus. Möglicherweise müssen Sie zuerst einen vorhandenen Cluster trennen, um die Dropdownliste für die Clusterauswahl zu aktivieren.

4. Wählen Sie Anfügen aus. Wenn der Cluster angehängt ist, sollte eine Erfolgsmeldung angezeigt werden.

Einen Amazon EMR in EKS-Cluster an einen EMR-Studio-Workspace anhängen

Zusätzlich zur Verwendung von Amazon EMR-Clustern, die auf Amazon ausgeführt werden EC2, können Sie einen Workspace an einen Amazon EMR on EKS-Cluster anhängen, um Notebook-Code auszuführen. Weitere Informationen zu Amazon EMR in EKS finden Sie unter [Was ist Amazon EMR in EKS](#).

Bevor Sie einen Workspace mit einem Amazon EMR in EKS-Cluster verbinden können, muss Ihnen Ihr Studio-Administrator Zugriffsberechtigungen erteilen.

Note

Sie können keinen Cluster von Amazon EMR in EKS in einem EMR Studio starten, das die Verbreitung vertrauenswürdiger Identitäten von IAM Identity Center verwendet.

On create

So fügen Sie beim Erstellen eines Workspace einen Amazon EMR in EKS-Cluster an

1. Erweitern Sie im Dialogfeld Workspace erstellen den Abschnitt Erweiterte Konfiguration.
2. Wählen Sie Workspace an einen Amazon EMR in EKS-Cluster anfügen.
3. Wählen Sie unter Amazon EMR in EKS-Cluster einen Cluster aus der Dropdownliste aus.
4. Wählen Sie unter Endpunkt auswählen einen verwalteten Endpunkt aus, der an den Workspace angefügt werden soll. Ein verwalteter Endpunkt ist ein Gateway, über das EMR Studio mit dem von Ihnen ausgewählten Cluster kommunizieren kann.
5. Wählen Sie Workspace erstellen aus, um den Workspace-Erstellungsprozess abzuschließen und den ausgewählten Cluster anzuhängen.

Nachdem Sie einen Cluster angehängt haben, können Sie den Workspace-Erstellungsprozess abschließen. Wenn Sie den neuen Workspace zum ersten Mal öffnen und den Bereich EMR-Cluster auswählen, sollte Ihr ausgewählter Cluster angehängt sein.

In the Workspace UI

So hängen Sie über die Workspace-Benutzeroberfläche einen Amazon EMR an einen EKS-Cluster an

1. Wählen Sie in dem Workspace, den Sie einem Cluster zuordnen möchten, in der linken Seitenleiste das EMR-Cluster-Symbol aus, um das Cluster-Bereich zu öffnen.
2. Erweitern Sie die Dropdownliste Clustertyp und wählen Sie EMR-Cluster in EKS aus.
3. Wählen Sie unter EMR in EKS-Cluster einen Cluster aus der Dropdownliste aus.
4. Wählen Sie unter Endpunkt einen verwalteten Endpunkt aus, der an den Workspace angehängt werden soll. Ein verwalteter Endpunkt ist ein Gateway, über das EMR Studio mit dem von Ihnen ausgewählten Cluster kommunizieren kann.
5. Wählen Sie Anfügen aus. Wenn der Cluster angehängt ist, sollte eine Erfolgsmeldung angezeigt werden.

Eine Amazon-EMR-Serverless-Anwendung an einen EMR Studio Workspace anhängen

Sie können einen Workspace an eine EMR-Serverless-Anwendung anhängen, um interaktive Workloads auszuführen. Weitere Informationen finden Sie unter [Verwenden von Notebooks zur Ausführung interaktiver Workloads mit EMR Serverless über EMR Studio](#).

Note

Sie können eine EMR-Serverless-Anwendung nicht an ein EMR Studio anhängen, das die Verbreitung vertrauenswürdiger Identitäten von IAM Identity Center Trusted verwendet.

Example Einen Workspace an eine serverlose EMR-Anwendung anhängen in JupyterLab

Bevor Sie einen Workspace mit einer EMR-Serverless-Anwendung verbinden können, muss Ihnen Ihr Kontoadministrator Zugriffsberechtigungen gewähren, wie unter [Erforderliche Berechtigungen für interaktive Workloads](#) beschrieben.

1. Navigieren Sie zu EMR Studio, wählen Sie Ihren Workspace aus und wählen Sie dann Workspace starten > Schnellstart aus.
2. Öffnen Sie in JupyterLab der linken Seitenleiste den Tab Cluster.

3. Wählen Sie EMR Serverless als Rechenoption aus, wählen Sie dann eine EMR-Serverless-Anwendung und eine Laufzeit-Rolle aus.
4. Wählen Sie Anfügen, um den Cluster an Ihren Workspace anzufügen.

Wenn Sie jetzt diesen Workspace öffnen, sollten Sie sehen, dass Ihre ausgewählte Anwendung angefügt ist.

Einen neuen EMR-Cluster erstellen und an einen EMR Studio Workspace anhängen

Fortgeschrittene EMR Studio-Benutzer können neue EMR-Cluster bereitstellen, die auf Amazon laufen, EC2 um sie mit einem Workspace zu verwenden. Auf dem neuen Cluster sind standardmäßig alle Big-Data-Anwendungen installiert, die für EMR Studio erforderlich sind.

Um Cluster zu erstellen, muss Ihnen Ihr Studio-Administrator zunächst mithilfe einer Sitzungsrichtlinie die Erlaubnis erteilen. Weitere Informationen finden Sie unter [Berechtigungsrichtlinien für EMR-Studio-Benutzer erstellen](#).

Sie können einen neuen Cluster im Dialogfeld Workspace erstellen oder im Bereich Cluster in der Workspace-Benutzeroberfläche erstellen. In beiden Fällen haben Sie zwei Möglichkeiten zum Erstellen eines Clusters:

1. Einen EMR-Cluster erstellen — Erstellen Sie einen EMR-Cluster, indem Sie den EC2 Amazon-Instance-Typ und die Anzahl auswählen.
2. Eine Cluster-Vorlage verwenden – Stellen Sie einen Cluster bereit, indem Sie eine vordefinierte Cluster-Vorlage auswählen. Diese Option wird angezeigt, wenn Sie berechtigt sind, Clustervorlagen zu verwenden.

Note

Wenn Sie die Verbreitung vertrauenswürdiger Identitäten von IAM Identity Center für Ihr Studio aktiviert haben, müssen Sie eine Vorlage verwenden, um einen Cluster zu erstellen.

So erstellen Sie einen EMR-Cluster durch Bereitstellung einer Clusterkonfiguration

1. Wählen Sie einen Startpunkt aus.

Zu ...	Vorgehensweise
Erstellen Sie den Cluster, wenn Sie einen Workspace mit dem Dialogfeld Workspace erstellen.	Erweitern Sie den Abschnitt Erweiterte Konfiguration im Dialogfeld Workspace erstellen und wählen Sie EMR-Cluster erstellen aus.
Erstellen Sie den Cluster über das EMR-Cluster-Panel in der Workspace-Benutzer oberfläche, nachdem Sie einen Workspace erstellt haben.	Wählen Sie in der linken Seitenleiste eines geöffneten Workspace die Registerkarte EMR-Cluster, erweitern Sie den Abschnitt Erweiterte Konfiguration und wählen Sie Cluster erstellen aus.

2. Geben Sie einen Clusternamen ein. Wenn Sie den Cluster benennen, können Sie ihn später in der Liste der EMR-Studio-Cluster leichter finden.
3. Wählen Sie für die Amazon-EMR-Version eine Amazon-EMR-Release-Version für den Cluster aus.
4. Wählen Sie unter Instance den Typ und die Anzahl der EC2 Amazon-Instances für den Cluster aus. Weitere Informationen zur Auswahl von Instance-Typen finden Sie unter [EC2 Amazon-Instance-Typen für die Verwendung mit Amazon EMR konfigurieren](#). Genau eine Instance wird als Primärknoten verwendet.
5. Wählen Sie ein Subnetz aus, in dem EMR Studio den neuen Cluster starten kann. Jede Subnetzoption wurde von Ihrem Studio-Administrator vorab genehmigt, und Ihr Workspace sollte in der Lage sein, eine Verbindung zu einem Cluster in einem beliebigen aufgelisteten Subnetz herzustellen.
6. Wählen Sie eine S3-URI für die Protokollspeicherung.
7. Wählen Sie EMR-Cluster erstellen aus, um den Cluster zu bereitzustellen. Wenn Sie das Dialogfeld Workspace erstellen verwenden, wählen Sie Workspace erstellen aus, um den Workspace zu erstellen und den Cluster bereitzustellen. Nachdem EMR Studio den neuen Cluster bereitgestellt hat, wird der Cluster an den Workspace angehängt.

So erstellen Sie einen Cluster mit einer Cluster-Vorlage

1. Wählen Sie einen Startpunkt aus.

Zu ...	Vorgehensweise
Erstellen Sie den Cluster, wenn Sie einen Workspace mit dem Dialogfeld Workspace erstellen.	Erweitern Sie den Abschnitt Erweiterte Konfiguration im Dialogfeld Workspace erstellen und wählen Sie Cluster-Vorlage verwenden aus.
Erstellen Sie den Cluster über das EMR-Cluster-Panel in der Workspace-Benutzeroberfläche.	Wählen Sie in der linken Seitenleiste eines geöffneten Workspace die Registerkarte EMR-Cluster, erweitern Sie den Abschnitt Erweiterte Konfiguration und wählen Sie Cluster-Vorlage aus.

2. Wählen Sie eine Cluster-Vorlage aus der Dropdown-Liste aus. Jede verfügbare Clustervorlage enthält eine kurze Beschreibung, die Ihnen bei der Auswahl hilft.
3. Die von Ihnen gewählte Cluster-Vorlage kann zusätzliche Parameter wie die Amazon-EMR-Release-Version oder den Clusternamen enthalten. Sie können Werte auswählen oder einfügen oder die Standardwerte verwenden, die Ihr Administrator ausgewählt hat.
4. Wählen Sie ein Subnetz aus, in dem EMR Studio den neuen Cluster starten kann. Jede Subnetzoption wurde von Ihrem Studio-Administrator vorab genehmigt, und Ihr Workspace sollte in der Lage sein, eine Verbindung zu einem Cluster in einem beliebigen Subnetz herzustellen.
5. Wählen Sie Clustervorlage verwenden, um den Cluster bereitzustellen und an den Workspace anzuhängen. Es dauert einige Minuten, bis EMR Studio den Cluster erstellt. Wenn Sie das Dialogfeld Workspace erstellen verwenden, wählen Sie Workspace erstellen aus, um den Workspace zu erstellen und den Cluster bereitzustellen. Nachdem EMR Studio den neuen Cluster bereitgestellt hat, wird der Cluster an den Workspace angehängt.

Trennen Sie einen Computer von einem EMR Studio Workspace

Um den mit einem Workspace verbundenen Cluster auszutauschen, können Sie einen Cluster von der Workspace-Benutzeroberfläche trennen.

So trennen Sie einen Cluster von einem Workspace

1. Wählen Sie in dem Workspace, den Sie die Zuordnung zu einem Cluster aufheben möchten, in der linken Seitenleiste das EMR-Cluster-Symbol aus, um das Cluster-Bereich zu öffnen.

2. Wählen Sie unter Cluster auswählen die Option Trennen aus und warten Sie, bis EMR Studio den Cluster getrennt hat. Wenn der Cluster getrennt ist, sehen Sie eine Erfolgsmeldung.

So trennen Sie eine Serverless-EMR-Anwendung von einem EMR-Studio-Workspace

Um den mit einem Workspace verbundenen Compute auszutauschen, können Sie eine Anwendung von der Workspace-Benutzeroberfläche trennen.

1. Wählen Sie in dem Workspace, den Sie von einem Cluster trennen möchten, in der linken Seitenleiste das Amazon-EMR-Datenverarbeitungssymbol aus, um das Datenverarbeitungs-Panel zu öffnen.
2. Wählen Sie unter Compute auswählen die Option Trennen aus und warten Sie, bis EMR Studio die Anwendung getrennt hat. Wenn die Anwendung getrennt ist, sehen Sie eine Erfolgsmeldung.

Git-basierte Repositorys mit einem EMR Studio Workspace verknüpfen

Ordnen Sie einem Amazon EMR Studio Workspace bis zu drei Git-basierte Repositorys zu, um Notebookdateien zu speichern und gemeinsam zu nutzen.

Über Git-Repositorys für EMR Studio

Sie können einem EMR Studio Workspace maximal drei Git-Repositorys zuordnen. Standardmäßig können Sie in jedem Workspace aus einer Liste von Git-Repositorys wählen, die demselben AWS Konto wie das Studio zugeordnet sind. Sie können auch ein neues Git-Repository als Ressource für einen Workspace erstellen.

Sie können Git-Befehle wie die folgenden mit einem Terminalbefehl ausführen, während Sie mit dem Primärknoten eines Clusters verbunden sind.

```
!git pull origin <branch-name>
```

Alternativ kannst du den jupyterlab-git Erweiterung. Öffnen Sie es in der linken Seitenleiste, indem Sie das Git-Symbol auswählen. [Informationen zur Jupyterlab-Git-Erweiterung für finden Sie unter jupyterlab-git. JupyterLab](#)

Voraussetzungen

- Um ein Git-Repository mit einem Workspace zu verknüpfen, muss Ihr Studio so konfiguriert sein, dass die Verknüpfung mit Git-Repositorys zulässig ist. Ihr Studio-Administrator sollte folgende Schritte unternehmen, um [Zugriff und Berechtigungen für Git-basierte Repositorys einrichten](#).
- Wenn Sie ein CodeCommit Repository verwenden, müssen Sie Git-Anmeldeinformationen und HTTPS verwenden. SSH-Schlüssel und HTTPS mit dem AWS Command Line Interface Credential Helper werden nicht unterstützt. CodeCommit unterstützt auch keine persönlichen Zugriffstoken (PATs). Weitere Informationen finden Sie unter [Using IAM with CodeCommit](#) im IAM-Benutzerhandbuch und [Setup für HTTPS-Benutzer mit Git-Anmeldeinformationen](#) im AWS CodeCommit User Guide.

Anweisungen

So verknüpfen Sie ein zugeordnetes Git-Repository mit einem Workspace

1. Öffnen Sie den Workspace, den Sie mit einem Repository verknüpfen möchten, in der Workspaces-Liste im Studio.
2. Wählen Sie in der linken Seitenleiste das Amazon-EMR-Git-Repository-Symbol, um das Git-Repository-Toolpanel zu öffnen.
3. Erweitern Sie unter Git-Repositorys die Drop-down-Liste und wählen Sie maximal drei Repositorys aus, die mit dem Workspace verknüpft werden sollen. EMR Studio registriert Ihre Auswahl und beginnt, jedes Repository zu verknüpfen.

Es kann einige Zeit dauern, bis der Verbindungsvorgang abgeschlossen ist. Sie können den Status für jedes Repository sehen, das Sie im Git-Repository-Toolpanel ausgewählt haben. Nachdem EMR Studio ein Repository mit einem Workspace verknüpft hat, sollten die Dateien, die zu diesem Repository gehören, im Dateibrowser-Bereich angezeigt werden.

Um einem Workspace ein neues Git-Repository als Ressource hinzuzufügen

1. Öffnen Sie den Workspace, den Sie mit einem Repository verknüpfen möchten, in der Workspaces-Liste im Studio.
2. Wählen Sie in der linken Seitenleiste das Amazon-EMR-Git-Repository-Symbol, um das Git-Repository-Toolpanel zu öffnen.
3. Wählen Sie Neues Git-Repository hinzufügen.

4. Geben Sie unter Repository-Name einen Namen ein, der für das Repository in EMR Studio verwendet werden soll. Namen dürfen nur alphanumerische Zeichen, Bindestriche oder Unterstriche enthalten.
5. Geben Sie für Git repository URL (Git-Repository-URL) die URL für das Repository ein. Wenn Sie ein CodeCommit Repository verwenden, wird diese URL kopiert, wenn Sie „URL klonen“ und dann „HTTPS klonen“ wählen. Beispiel, `https://git-codecommit.us-west-2.amazonaws.com/v1/repos/[MyCodeCommitRepoName]`.
6. Geben Sie für Branch den Namen eines vorhandenen Branches ein, den Sie auschecken möchten.
7. Wählen Sie Optionen für Git-Anmeldeinformationen gemäß den folgenden Richtlinien. EMR Studio greift mithilfe von Geheimnissen, die im Secrets Manager gespeichert sind, auf Ihre Git-Anmeldeinformationen zu.

 Note

Wenn Sie ein GitHub Repository verwenden, empfehlen wir Ihnen, zur Authentifizierung ein Personal Access Token (PAT) zu verwenden. Ab dem 13. August 2021 ist eine tokenbasierte Authentifizierung erforderlich und bei der Authentifizierung von Git-Vorgängen werden keine Passwörter mehr akzeptiert. GitHub Weitere Informationen finden Sie im Beitrag [Token-Authentifizierungsanforderungen für Git-Operationen](#) im GitHub Blog.

Option	Beschreibung
Erstellen eines neuen Secrets	Wählen Sie diese Option, um bestehende Git-Anmeldeinformationen mit einem neuen Geheimnis zu verknüpfen, das AWS Secrets Manager für Sie erstellt wird. Führen Sie basierend auf den Git-Anmeldeinformationen, die Sie für das Repository verwenden, einen der folgenden Schritte aus. Wenn Sie für den Zugriff auf das Repository einen Git-Benutzernamen mit Passwort verwenden, wählen Sie Benutzername

Option	Beschreibung
	und Passwort aus, geben Sie den Namen des Secrets ein, das in Secrets Manager verwendet werden soll, und geben Sie dann den Benutzernamen und das Passwort ein, die mit dem Secret verknüpft werden sollen. -ODER- Wenn Sie ein persönliches Zugriffstoken für den Zugriff auf das Repository verwenden , wählen Sie Persönliches Zugriffstoken (PAT) aus, geben Sie den Secret-Name ein, der in Secrets Manager verwendet werden soll, und geben Sie dann Ihr persönliches Zugriffstoken ein. Weitere Informationen findest du unter Persönliches Zugriffstoken für die Befehlszeile erstellen für GitHub und Persönliche Zugriffstoken für Bitbucket. CodeCommit Repositorys unterstützen diese Option nicht.
Verwenden eines öffentlichen Repository ohne Anmeldeinformationen	Wählen Sie diese Option, um auf ein öffentliches Repository zuzugreifen.
Verwenden Sie ein vorhandenes Geheimnis AWS	Wählen Sie diese Option, wenn Sie Ihre Anmeldeinformationen bereits als Secret in Secrets Manager gespeichert haben, und wählen Sie dann den Namen des Secrets in der Liste aus. Wenn Sie ein Secret auswählen, das mit einem Git-Benutzernamen und -Passwort verknüpft ist, muss das Secret das Format <code>{"gitUsername": " <i>MyUserName</i> ", "gitPassword": " <i>MyPassword</i> "}</code> aufweisen.

8. Wählen Sie Repository hinzufügen, um das neue Repository zu erstellen. Nachdem EMR Studio das neue Repository erstellt hat, wird eine Erfolgsmeldung angezeigt. Das neue Repository erscheint in der Dropdown-Liste unter Git-Repositorys.
9. Um das neue Repository mit deinem Workspace zu verknüpfen, wähle es aus der Drop-down-Liste unter Git-Repositorys aus.

Es kann einige Zeit dauern, bis der Verbindungsvorgang abgeschlossen ist. Nachdem EMR Studio das neue Repository mit dem Workspace verknüpft hat, sollte im Dateibrowser-Bereich ein neuer Ordner mit demselben Namen wie Ihr Repository angezeigt werden.

Um ein anderes verknüpftes Repository zu öffnen, navigieren Sie im Dateibrowser zu seinem Ordner.

Verwenden des Editors von Amazon Athena SQL in EMR Studio

Übersicht

Sie können Amazon EMR Studio verwenden, um interaktive Abfragen in Amazon Athena zu entwickeln und auszuführen. Das bedeutet, dass Sie SQL-Analysen in Athena von derselben EMR-Studio-Oberfläche aus durchführen können, mit der Sie Ihre Spark-, Scala- und andere Workloads ausführen. Mit dieser Integration können Sie die automatische Vervollständigung verwenden, um schnell Abfragen zu entwickeln, Daten in Ihrem AWS Glue-Datenkatalog zu durchsuchen, gespeicherte Abfragen zu erstellen, Ihren Abfrageverlauf einzusehen und vieles mehr.

Weitere Informationen finden Sie unter [Verwenden von Athena SQL](#) im Amazon-Athena-Benutzerhandbuch.

Den Athena-SQL-Editor in EMR Studio verwenden

Gehen Sie wie folgt vor, um interaktive Abfragen in Amazon Athena von Ihrem EMR Studio aus zu entwickeln und auszuführen:

1. Fügen Sie der Benutzerrolle die erforderlichen Berechtigungen für die Benutzer hinzu, die auf die Workspaces in diesem Studio zugreifen. Die Berechtigungen sind in der [AWS Identity and Access Management Berechtigungen für EMR Studio-Benutzer](#)-Tabelle in der Spalte Access Amazon Athena SQL editor from your EMR Studio (Zugriff auf den Editor von Amazon Athena SQL von Ihrem EMR Studio aus) aufgeführt. Alternativ können Sie den Inhalt der erweiterten Richtlinie aus [Beispielbenutzerrichtlinien](#) kopieren, um Benutzern vollständige Berechtigungen für die Funktionen von EMR Studio zu gewähren, einschließlich dieser.

2. [Richten Sie ein EMR Studio ein](#) und [erstellen Sie es](#).
3. Navigieren Sie zu Ihrem Studio und wählen Sie in der Seitenleiste den Query Editor aus.

Sie sollten jetzt die bekannte Athena-Editor-Benutzeroberfläche sehen. Informationen zu den ersten Schritten und zur Verwendung von Athena SQL zur Ausführung interaktiver Abfragen finden Sie unter [Erste Schritte](#) und [Verwendung von Athena SQL](#) im Benutzerhandbuch zu Amazon Athena.

Note

Wenn Sie die Verbreitung vertrauenswürdiger Identitäten über IAM Identity Center für Ihr EMR Studio aktiviert haben, müssen Sie Athena-Arbeitsgruppen verwenden, um den Abfragezugriff zu steuern, und die Arbeitsgruppe, die Sie verwenden, muss auch die vertrauenswürdige Identitätsverbreitung verwenden. Schritte zur Einrichtung von Identity Center und zur Aktivierung der Verbreitung vertrauenswürdiger Identitäten für Ihre Arbeitsgruppe finden Sie unter [Verwenden von IAM-Identity-Center-fähigen Athena-Arbeitsgruppen](#) im Amazon-Athena-Benutzerhandbuch.

Überlegungen zur Verwendung des Athena-SQL-Editors in EMR Studio

- Die Integration in Athena ist in allen kommerziellen Regionen verfügbar, in denen EMR Studio und Athena verfügbar sind.
- Die folgenden Athena-Features sind in EMR Studio nicht verfügbar:
 - Admin-Features wie das Erstellen oder Aktualisieren von Athena-Arbeitsgruppen, Datenquellen oder Kapazitätsreservierungen
 - Athena-for-Spark- oder Spark-Notebooks
 - DataZone Amazon-Integration
 - Kostenbasierter Optimierer (CBO)
 - Step Functions

CodeWhisperer Amazon-Integration mit EMR Studio Workspaces

Übersicht

Sie können [Amazon CodeWhisperer mit Amazon](#) EMR Studio verwenden, um Empfehlungen in Echtzeit zu erhalten, während Sie Code JupyterLab einschreiben. CodeWhisperer kann Ihre

Kommentare vervollständigen, einzelne Codezeilen fertigstellen, line-by-line Empfehlungen aussprechen und vollständig formatierte Funktionen generieren.

Note

Wenn Sie Amazon EMR Studio verwenden, werden AWS möglicherweise Daten über Ihre Nutzung und Inhalte gespeichert, um den Service zu verbessern. Weitere Informationen und Anweisungen zum Deaktivieren der [Datenweitergabe finden Sie unter Teilen Ihrer Daten mit AWS](#) im CodeWhisperer Amazon-Benutzerhandbuch.

Überlegungen zur Verwendung CodeWhisperer mit Workspaces

- CodeWhisperer Die Integration ist dort verfügbar, AWS-Regionen wo EMR Studio verfügbar ist, wie in den Überlegungen zu [EMR Studio](#) dokumentiert.
- Amazon EMR Studio verwendet automatisch den CodeWhisperer Endpunkt in USA Ost (Nord-Virginia) (us-east-1) für Empfehlungen, unabhängig von der Region, in der sich Ihr Studio befindet.
- CodeWhisperer unterstützt nur die Python-Sprache für die Codierung von ETL-Skripts für Spark-Jobs in EMR Studio.
- Eine clientseitige Telemetrieoption quantifiziert Ihre Nutzung von. CodeWhisperer Diese Funktion wird mit EMR Studio nicht unterstützt.

Berechtigungen erforderlich für CodeWhisperer

Zur Verwendung CodeWhisperer müssen Sie Ihrer IAM-Benutzerrolle für Amazon EMR Studio die folgende Richtlinie hinzufügen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeWhispererPermissions",
      "Effect": "Allow",
      "Action": [ "codewhisperer:GenerateRecommendations" ],
      "Resource": "*"
    }
  ]
}
```

Mit Workspaces verwenden CodeWhisperer

Um das CodeWhisperer Referenzprotokoll anzuzeigen JupyterLab, öffnen Sie das CodeWhispererFenster am unteren Rand des JupyterLab Fensters und wählen Sie „Code-Referenzprotokoll öffnen“.

Die folgende Liste enthält Tastenkombinationen, mit denen Sie mit CodeWhisperer Vorschlägen interagieren können:

- Empfehlungen pausieren — Verwende die Option Automatische Vorschläge pausieren in den CodeWhisperer Einstellungen.
- Eine Empfehlung akzeptieren – Drücken Sie die Tabulatortaste auf Ihrer Tastatur.
- Eine Empfehlung ablehnen – Drücken Sie die Escape-Taste auf Ihrer Tastatur.
- Empfehlungen durchsuchen – Verwenden Sie die Aufwärts- und Abwärtspfeile auf Ihrer Tastatur.
- Manueller Aufruf – Drücken Sie die Tasten Alt und C auf Ihrer Tastatur. Wenn Sie einen Mac verwenden, drücken Sie Cmd und C.

Sie können sie auch verwenden CodeWhisperer , um Einstellungen wie die Protokollebene zu ändern und Vorschläge für Codereferenzen zu erhalten. Weitere Informationen finden Sie unter [Einrichtung CodeWhisperer mit JupyterLab](#) und [Funktionen](#) im CodeWhisperer Amazon-Benutzerhandbuch.

Debuggen von Anwendungen und Aufträgen mit EMR Studio

Mit Amazon EMR Studio können Sie Datenanwendungsschnittstellen starten, um Anwendungen und Auftragsausführungen im Browser zu analysieren.

Sie können die persistenten Benutzeroberflächen außerhalb des Clusters für Amazon EMR, die auf EC2 Clustern ausgeführt werden, auch von der Amazon EMR-Konsole aus starten. Weitere Informationen finden Sie unter [Persistente Anwendungsbenuzeroberflächen in Amazon EMR anzeigen](#).

Note

Abhängig von Ihren Browsereinstellungen müssen Sie möglicherweise Popups aktivieren, damit die Benutzeroberfläche einer Anwendung geöffnet werden kann.

Informationen zum Konfigurieren und Verwenden der Anwendungsschnittstellen finden Sie unter [YARN Timeline Server](#), [Überwachung und Instrumentierung](#) oder [Tez-UI-Übersicht](#).

Debuggen Sie Amazon EMR, das auf Amazon-Jobs ausgeführt wird EC2

Workspace UI

Starten Sie eine Cluster-Benutzeroberfläche aus einer Notebook-Datei

Wenn Sie die Amazon-EMR-Release-Versionen 5.33.0 und höher verwenden, können Sie die Spark-Webbenutzeroberfläche (die Spark-Benutzeroberfläche oder den Spark History Server) von einem Notebook in Ihrem Workspace aus starten.

Uls Arbeiten Sie auf dem Cluster mit den PySpark Kernen, Spark oder SparkR. Die maximale sichtbare Dateigröße für Spark-Event- oder Container-Logs beträgt 10 MB. Wenn Ihre Protokolldateien 10 MB überschreiten, empfehlen wir Ihnen, zum Debuggen von Jobs den persistenten Spark History Server anstelle der Cluster-internen Spark-Benutzeroberfläche zu verwenden.

Important

Damit EMR Studio Benutzeroberflächen für Cluster-Anwendungen von einem Workspace aus starten kann, muss ein Cluster in der Lage sein, mit dem Amazon API Gateway zu kommunizieren. Sie müssen den EMR-Cluster so konfigurieren, dass ausgehender Netzwerkverkehr zu Amazon API Gateway zugelassen wird, und sicherstellen, dass Amazon API Gateway vom Cluster aus erreichbar ist.

Die Spark-Benutzeroberfläche greift auf Container-Logs zu, indem sie Hostnamen auflöst. Wenn Sie einen benutzerdefinierten Domainnamen verwenden, müssen Sie sicherstellen, dass die Hostnamen Ihrer Clusterknoten von Amazon DNS oder dem von Ihnen angegebenen DNS-Server aufgelöst werden können. Stellen Sie dazu die Dynamic Host Configuration Protocol (DHCP)-Optionen für die Amazon Virtual Private Cloud (VPC) ein, die Ihrem Cluster zugeordnet ist. Weitere Informationen zu DHCP-Optionen finden Sie unter [DHCP-Optionssätze](#) im Amazon-Virtual-Private-Cloud-Benutzerhandbuch.

1. Öffnen Sie in Ihrem EMR Studio den Workspace, den Sie verwenden möchten, und stellen Sie sicher, dass er mit einem Amazon EMR-Cluster verbunden ist, auf dem ausgeführt wird. EC2 Detaillierte Anweisungen finden Sie unter [Einen Computer an einen EMR Studio Workspace anhängen](#).

- Öffnen Sie eine Notebook-Datei und verwenden Sie den PySpark, Spark- oder SparkR-Kernel. Um einen Kernel auszuwählen, wählen Sie den Kernel-Namen oben rechts in der Notebook-Symbolleiste, um das Dialogfeld Kernel auswählen zu öffnen. Der Name erscheint als Kein Kernel! wenn kein Kernel ausgewählt wurde.
- Führen Sie Ihren Notebook-Code aus. Folgendes wird als Ausgabe im Notebook angezeigt, wenn Sie den Spark-Kontext starten. Es kann einige Sekunden dauern, bis es angezeigt wird. Wenn Sie den Spark-Kontext gestartet haben, können Sie den %%info-Befehl ausführen, um jederzeit auf einen Link zur Spark-Benutzeroberfläche zuzugreifen.



Note

Wenn die Spark-UI-Links nicht funktionieren oder nach einigen Sekunden nicht angezeigt werden, erstellen Sie eine neue Notebook-Zelle und führen Sie den %%info-Befehl aus, um die Links neu zu generieren.

[1]:

sc

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
2	application_1613085840432_0003	spark	idle	Link	Link	✓

SparkSession available as 'spark'.

res1: org.apache.spark.SparkContext = org.apache.spark.SparkContext@58262802

- Um die Spark-Benutzeroberfläche zu starten, wählen Sie Link unter Spark-Benutzeroberfläche. Wenn Ihre Spark-Anwendung ausgeführt wird, wird die Spark-Benutzeroberfläche in einer neuen Registerkarte geöffnet. Wenn die Anwendung abgeschlossen ist, wird stattdessen der Spark History Server geöffnet.

Nachdem Sie die Spark-Benutzeroberfläche gestartet haben, können Sie die URL im Browser ändern, um den YARN ResourceManager - oder den Yarn Timeline Server zu öffnen. Fügen Sie nach `amazonaws.com` einen der folgenden Pfade hinzu.

Web-Benutzer- oberfläche	Pfad	Beispiel für eine geänderte URL
GARN ResourceM anager	/rm	https:// <i>j-examplebby5ij</i> .emrappui-prod. <i>eu-west-1</i> .amazonaws.com <i>/rm</i>
Yarn-Timeline- Server	/yts	https://.emrappui-prod. <i>j-examplebby5ij</i> <i>eu-west-1</i> .amazonaws.com <i>/yts</i>
Spark History Server	/shs	https://.emrappui-prod. <i>j-examplebby5ij</i> <i>eu-west-1</i> .amazonaws.com <i>/shs</i>

Studio UI

Starten Sie den persistenten YARN Timeline Server, Spark History Server oder Tez UI über die EMR-Studio-Benutzeroberfläche

1. Wählen Sie in Ihrem EMR Studio links EC2 auf der Seite Amazon EMR aus, um die Liste Amazon EMR on EC2 Clusters zu öffnen.
2. Filtern Sie die Clusterliste nach Name, Status oder ID, indem Sie Werte in das Suchfeld eingeben. Sie können auch nach Erstellungszeitraum suchen.
3. Wählen Sie einen Cluster aus und wählen Sie dann Anwendung starten UIs, um eine Anwendungsbenutzeroberfläche auszuwählen. Die Anwendungsbenutzeroberfläche wird in einer neuen Browser-Registerkarte geöffnet. Es kann einige Zeit dauern, bis sie geladen wird.

Debuggen Sie EMR Studio, das auf EMR Serverless läuft

Ähnlich wie Amazon EMR, das auf Amazon läuft EC2, können Sie die Workspace-Benutzeroberfläche verwenden, um Ihre EMR Serverless-Anwendungen zu analysieren. Wenn Sie Amazon-EMR-Versionen 6.14.0 und höher verwenden, können Sie von der Workspace-Benutzeroberfläche aus die Spark-Webbenutzeroberfläche (die Spark-Benutzeroberfläche oder den Spark History Server) von einem Notebook in Ihrem Workspace aus starten. Der Einfachheit halber stellen wir auch einen Link zum Treiberprotokoll zur Verfügung, über den Sie schnell auf die Spark-Treiberprotokolle zugreifen können.

Debuggen Sie Amazon EMR in EKS-Auftragsausführungen mit dem Spark History Server

Wenn Sie eine Auftragsausführung an einen Amazon EMR in EKS-Cluster senden, können Sie über den Spark History Server auf die Protokolle für diese Auftragsausführung zugreifen. Der Spark History Server bietet Tools zur Überwachung von Spark-Anwendungen, z. B. eine Liste von Scheduler-Phasen und -aufgaben, eine Zusammenfassung der RDD-Größen und der Speichernutzung sowie Umgebungsinformationen. Sie können den Spark History Server für Amazon EMR in EKS-Auftragsläufe auf folgende Weise starten:

- Wenn Sie einen Auftrag einreichen, der mit EMR Studio und einem von Amazon EMR auf EKS verwalteten Endpunkt ausgeführt wird, können Sie den Spark History Server von einer Notebook-Datei in Ihrem Workspace aus starten.
- Wenn Sie einen Job einreichen, der mit dem AWS CLI oder AWS SDK für Amazon EMR auf EKS ausgeführt wird, können Sie den Spark History Server über die EMR Studio-Benutzeroberfläche starten.

Informationen zur Verwendung des Spark History Servers finden Sie unter [Überwachung und Instrumentierung](#) in der Apache-Spark-Dokumentation. Weitere Informationen zu Auftragsausführungen finden Sie unter [Konzepte und Komponenten](#) im Entwicklerhandbuch zu Amazon EMR in EKS.

So starten Sie den Spark History Server aus einer Notebook-Datei in Ihrem EMR Studio Workspace

1. Öffnen Sie einen Workspace, der mit einem Amazon EMR in EKS-Cluster verbunden ist.
2. Wählen Sie Ihre Notebook-Datei aus und öffnen Sie sie im Workspace.
3. Wählen Sie oben in der Notebook-Datei die Spark-Benutzeroberfläche, um den persistenten Spark-Geschichtsserver in einer neuen Registerkarte zu öffnen.

So starten Sie den Spark History Server über die EMR-Studio-Benutzeroberfläche

Note

In der Jobliste in der EMR Studio-Benutzeroberfläche werden nur Jobausführungen angezeigt, die Sie mit dem AWS CLI oder AWS SDK für Amazon EMR auf EKS einreichen.

1. Wählen Sie in Ihrem EMR Studio links auf der Seite Amazon EMR in EKS aus.
2. Suchen Sie nach dem virtuellen Amazon EMR in EKS-Cluster, mit dem Sie Ihre Auftragsausführung eingereicht haben. Sie können die Liste der Cluster nach Status oder ID filtern, indem Sie Werte in das Suchfeld eingeben.
3. Wählen Sie den Cluster aus, um seine Detailseite zu öffnen. Auf der Detailseite werden Informationen über den Cluster wie ID, Namespace und Status angezeigt. Auf der Seite wird auch eine Liste aller Auftragsausführungen angezeigt, die an diesen Cluster übermittelt wurden.
4. Wählen Sie auf der Cluster-Detailseite einen Auftrag aus, der debuggt werden soll.
5. Wählen Sie oben rechts in der Auftragsliste die Option Spark History Server starten, um die Anwendungsoberfläche in einer neuen Browser-Registerkarte zu öffnen.

Kernel und Bibliotheken in einem EMR Studio Workspace installieren

Jedes Amazon EMR Studio Workspace wird mit einer Reihe vorinstallierter Bibliotheken und Kernel ausgeliefert.

Kernel und Bibliotheken auf Clustern, die auf Amazon laufen EC2

Sie können die Umgebung für EMR Studio auch auf folgende Weise anpassen, wenn Sie EMR-Cluster verwenden, die auf Amazon ausgeführt werden: EC2

- Jupyter-Notebook-Kernel und Python-Bibliotheken auf einem Cluster-Primärknoten installieren – Wenn Sie Bibliotheken mit dieser Option installieren, teilen sich alle Workspaces, die demselben Cluster zugeordnet sind, diese Bibliotheken gemeinsam. Sie können Kernel oder Bibliotheken von einer Notebook-Zelle aus installieren oder während Sie über SSH mit dem Primärknoten eines Clusters verbunden sind.
- Verwenden Sie Bibliotheken für Notebooks – Wenn Workspace-Benutzer Bibliotheken von einer Notebook-Zelle aus installieren und verwenden, sind diese Bibliotheken nur für dieses Notebook verfügbar. Mit dieser Option können verschiedene Notebooks, die denselben Cluster verwenden, arbeiten, ohne sich Gedanken über widersprüchliche Bibliotheksversionen machen zu müssen.

EMR Studio Workspaces haben dieselbe grundlegende Architektur wie EMR Notebooks. Sie können Jupyter-Notebook-Kernel und Python-Bibliotheken mit EMR Studio genauso installieren und verwenden wie mit EMR Notebooks. Detaillierte Anweisungen finden Sie unter [Installation und Verwendung von Kernen und Bibliotheken in EMR Studio](#).

Kernel und Bibliotheken in Amazon EMR in EKS-Clustern

Amazon EMR auf EKS-Clustern enthalten die Kernel PySpark und Python 3.7 mit einer Reihe vorinstallierter Bibliotheken. Amazon EMR in EKS unterstützt die Installation zusätzlicher Bibliotheken oder Cluster nicht.

Auf jedem Amazon EMR on EKS-Cluster sind das folgende Python und die folgenden PySpark Bibliotheken installiert:

- Python – boto3, cffi, future, ggplot, jupyter, kubernetes, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn
- PySpark – ggplot, jupyter, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn

Kernel und Bibliotheken für EMR-Serverless-Anwendungen

In jeder EMR Serverless-Anwendung sind das folgende Python und die folgenden PySpark Bibliotheken installiert:

- Python – ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn
- PySpark – ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn

Verbessern Sie Kernel mit magic Befehle in EMR Studio

Übersicht

Unterstützung für EMR Studio und EMR Notebooks magic Befehle. Magie Befehle, oder magics sind Erweiterungen, die der IPython Kernel bereitstellt, um Sie beim Ausführen und Analysieren von Daten zu unterstützen. IPython ist eine interaktive Shell-Umgebung, die mit Python erstellt wurde.

Amazon EMR unterstützt auch Sparkmagic, ein Paket, das Spark-bezogene Kernel (PySpark, SparkR- und Scala-Kernel) mit spezifischen magic Befehle und das Livy auf dem Cluster verwendet, um Spark-Jobs zu senden.

Sie können Folgendes verwenden ... magic Befehle, solange Sie einen Python-Kernel in Ihrem EMR-Notebook haben. In ähnlicher Weise unterstützt jeder SPARK-bezogene Kernel Sparkmagic Befehle.

Magic Befehle, auch genannt magics gibt es in zwei Varianten:

- Linie magics — Diese magic Befehle werden mit einem einzigen % Präfix gekennzeichnet und funktionieren mit einer einzigen Codezeile
- Zelle magics — Diese magic Befehle sind mit einem doppelten %% Präfix gekennzeichnet und funktionieren auf mehreren Codezeilen

Für alle verfügbaren magics, siehe [Auflisten magic and Sparkmagic commands](#).

Überlegungen und Einschränkungen

- EMR Serverless unterstützt die %%sh-Ausführung von spark-submit nicht. Die EMR Notebooks werden nicht unterstützt magic-Geräte.
- Amazon EMR auf EKS-Clustern unterstützt nicht Sparkmagic Befehle für EMR Studio. Das liegt daran, dass Spark-Kernel, die Sie mit verwalteten Endpunkten verwenden, in Kubernetes integriert sind und nicht unterstützt werden von Sparkmagic und Livy. Sie können die Spark-Konfiguration als Workaround direkt im SparkContext Objekt festlegen, wie das folgende Beispiel zeigt.

```
spark.conf.set("spark.driver.maxResultSize", '6g')
```

- Folgendes magic Befehle und Aktionen sind verboten durch AWS:
 - %alias
 - %alias_magic
 - %automagic
 - %macro
 - Ändern von proxy_user mit %configure
 - Ändern von KERNEL_USERNAME mit %env oder %set_env

Auflisten magic and Sparkmagic commands

Verwenden Sie die folgenden Befehle, um die verfügbaren aufzulisten magic Befehle:

- %lsmagic listet alle derzeit verfügbaren magic Funktionen.
- %%help listet derzeit verfügbare SPARK-bezogene magic Funktionen, die bereitgestellt werden von Sparkmagic Paket.

%%configure wird verwendet, um Spark zu konfigurieren

Eines der nützlichsten Sparkmagic commands ist der %%configure Befehl, der die Parameter für die Sitzungserstellung konfiguriert. Mithilfe von conf-Einstellungen können Sie jede Spark-Konfiguration konfigurieren, die in der [Konfigurationsdokumentation für Apache Spark](#) erwähnt wird.

Example Fügen Sie eine externe JAR-Datei aus dem Maven-Repository oder Amazon S3 zu EMR Notebooks hinzu

Sie können den folgenden Ansatz verwenden, um jedem SPARK-bezogenen Kernel, der unterstützt wird von Sparkmagic.

```
%%configure -f
{"conf": {
  "spark.jars.packages": "com.jsuereth:scala-arm_2.11:2.0,ml.combust.bundle:bundle-ml_2.11:0.13.0,com.databricks:dbutils-api_2.11:0.0.3",
  "spark.jars": "s3://amzn-s3-demo-bucket/my-jar.jar"
}}
```

Example : Konfigurieren von Hudi

Anschließend verwenden Sie den Notebook-Editor, um Ihr EMR Notebook für die Verwendung von Hudi zu konfigurieren.

```
%%configure
{ "conf": {
  "spark.jars": "hdfs://apps/hudi/lib/hudi-spark-bundle.jar,hdfs:///apps/hudi/lib/spark-spark-avro.jar",
  "spark.serializer": "org.apache.spark.serializer.KryoSerializer",
  "spark.sql.hive.convertMetastoreParquet":"false"
}}
```

%%sh verwenden, um **spark-submit** auszuführen

Die %%sh magic führt Shell-Befehle in einem Unterprozess auf einer Instanz Ihres verbundenen Clusters aus. In der Regel würden Sie einen der Spark-bezogenen Kernel verwenden, um Spark-Anwendungen auf Ihrem angeschlossenen Cluster auszuführen. Wenn Sie jedoch einen Python-Kernel verwenden möchten, um eine Spark-Anwendung einzureichen, können Sie Folgendes verwenden und ersetzt den Bucket-Namen durch Ihren Bucket-Namen in Kleinbuchstaben.

```
%%sh
spark-submit --master yarn --deploy-mode cluster s3://amzn-s3-demo-bucket/test.py
```

In diesem Beispiel benötigt der Cluster Zugriff auf den Speicherort von `s3://amzn-s3-demo-bucket/test.py`, andernfalls schlägt der Befehl fehl.

Sie können jeden Linux-Befehl mit dem `%%sh` magic. Wenn Sie Spark- oder YARN-Befehle ausführen möchten, verwenden Sie eine der folgenden Optionen, um einen `emr-notebook` Hadoop-Benutzer zu erstellen und dem Benutzer Berechtigungen zur Ausführung der Befehle zu gewähren:

- Sie können einen neuen Benutzer explizit erstellen, indem Sie die folgenden Befehle ausführen.

```
hadoop fs -mkdir /user/emr-notebook
hadoop fs -chown emr-notebook /user/emr-notebook
```

- Sie können den Benutzerwechsel in Livy aktivieren, wodurch der Benutzer automatisch erstellt wird. Weitere Informationen finden Sie unter [Aktivieren des Identitätswechsels zur Überwachung von Spark-Benutzer- und -Aufgabenaktivitäten](#).

Wird zur Visualisierung von `%%display`-Spark-Datenrahmen verwendet

Sie können den `%%display` magic um einen Spark-Datenrahmen zu visualisieren. Um das zu benutzen magic, führen Sie den folgenden Befehl aus.

```
%%display df
```

Wählen Sie, ob Sie die Ergebnisse in einem Tabellenformat anzeigen möchten, wie das folgende Bild zeigt.

Type:

Table

Pie

Scatter

Line

Area

Bar

year	month	total_passengers	total_trips
2012-01-01	3	26866837	16146923
2011-01-01	3	26091246	16066350
2013-01-01	3	26965079	15749228
2011-01-01	10	26287953	15707756
2009-01-01	10	26202049	15604551
2012-01-01	5	26278817	15567525
2011-01-01	5	25508952	15554868
2010-01-01	9	25533166	15540209
2010-01-01	5	26002858	15481351
2012-01-01	4	25900645	15477914

Sie können sich auch dafür entscheiden, Ihre Daten mit fünf Arten von Diagrammen zu visualisieren. Zu Ihren Optionen gehören Kreis-, Streu-, Linien-, Flächen- und Balkendiagramme.

Type:

Table

Pie

Scatter

Line

Area

Bar

Encoding:

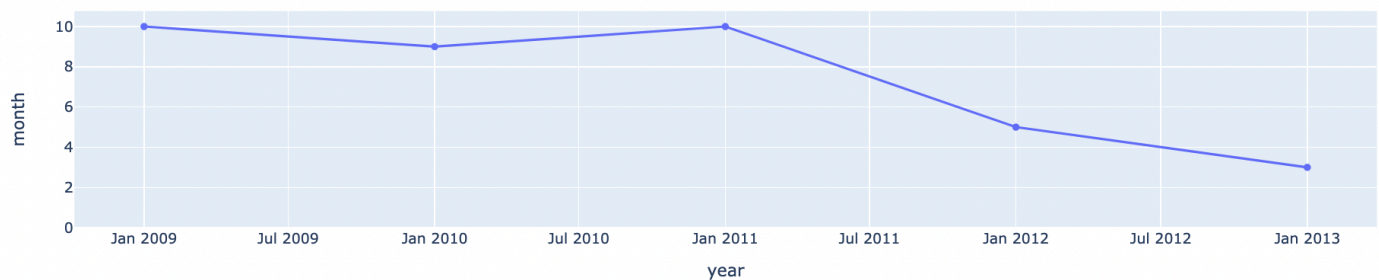
X year

Y month

Func. Max

Log scale X

Log scale Y



Verwenden Sie EMR Notebooks magicS

Amazon EMR bietet die folgenden EMR Notebooks magics, die Sie mit Python3- und Spark-basierten Kernen verwenden können:

- `%mount_workspace_dir` – Hängt Ihr Workspace-Verzeichnis in Ihren Cluster ein, sodass Sie Code aus anderen Dateien in Ihrem Workspace importieren und ausführen können

Note

Mit `%mount_workspace_dir` kann nur der Python-3-Kernel auf Ihre lokalen Dateisysteme zugreifen. Spark-Executoren haben mit diesem Kernel keinen Zugriff auf das bereitgestellte Verzeichnis.

- `%umount_workspace_dir` – Hängt Ihr Workspace-Verzeichnis von Ihrem Cluster ab
- `%generate_s3_download_url` – Generiert einen temporären Download-Link in Ihrer Notebook-Ausgabe für ein Amazon-S3-Objekt

Voraussetzungen

Vor der Installation von EMR Notebooks magics, führen Sie die folgenden Aufgaben aus:

- Stellen Sie sicher, dass [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#) Lesezugriff für Amazon S3 hat. Die `EMR_EC2_DefaultRole` mit der `AmazonElasticMapReduceforEC2Role` verwalteten Richtlinie erfüllt diese Anforderung. Wenn Sie eine benutzerdefinierte Rolle oder Richtlinie verwenden, stellen Sie sicher, dass sie über die erforderlichen S3-Berechtigungen verfügt.

Note

EMR Notebooks magics wird auf einem Cluster als Notebook-Benutzer ausgeführt und verwendet das EC2 Instance-Profil, um mit Amazon S3 zu interagieren. Wenn Sie ein Workspace-Verzeichnis auf einem EMR-Cluster mounten, können alle Workspaces und EMR-Notebooks, die berechtigt sind, eine Verbindung zu diesem Cluster herzustellen, auf das bereitgestellte Verzeichnis zugreifen.

Verzeichnisse werden standardmäßig schreibgeschützt bereitgestellt. Während `s3fs`-`fuse` und `goofys` Lese-/Schreibzugriffe ermöglichen, empfehlen wir dringend, die Bereitstellungsparameter nicht zu ändern, um Verzeichnisse im Lese-/Schreibmodus

bereitzustellen. Wenn Sie Schreibzugriff zulassen, werden alle am Verzeichnis vorgenommenen Änderungen in den S3-Bucket geschrieben. Um ein versehentliches Löschen oder Überschreiben zu vermeiden, können Sie die Versionsverwaltung für Ihren S3-Bucket aktivieren. Weitere Informationen finden Sie unter [Verwenden der Versionsverwaltung in S3-Buckets](#).

- Führen Sie eines der folgenden Skripts auf Ihrem Cluster aus, um die Abhängigkeiten für EMR Notebooks zu installieren magic-Geräte. Um ein Skript auszuführen, können Sie [Benutzerdefinierte Bootstrap-Aktionen verwenden](#) entweder den Anweisungen unter [Befehle und Skripts auf einem Amazon-EMR-Cluster ausführen](#) folgen, wenn Sie bereits über einen laufenden Cluster verfügen.

Sie können wählen, welche Abhängigkeit installiert werden soll. Sowohl [s3fs-fuse](#) als auch [goofys sind FUSE-Tools](#) (Filesystem in Userspace), mit denen Sie einen Amazon-S3-Bucket als lokales Dateisystem auf einem Cluster mounten können. Das s3fs-Tool bietet eine ähnliche Benutzererfahrung wie POSIX. Das goofys-Tool ist eine gute Wahl, wenn Sie Leistung einem POSIX-kompatiblen Dateisystem vorziehen.

Die Amazon EMR 7.x-Serie verwendet Amazon Linux 2023, das keine EPEL-Repositorys unterstützt. Wenn Sie Amazon EMR 7.x ausführen, folgen Sie zur Installation den [GitHubs3fs-fuse-Anweisungen](#). s3fs-fuse Wenn Sie die Serie 5.x oder 6.x verwenden, verwenden Sie zur Installation die folgenden Befehle. s3fs-fuse

```
#!/bin/sh

# Install the s3fs dependency for EMR Notebooks magics
sudo amazon-linux-extras install epel -y
sudo yum install s3fs-fuse -y
```

ODER

```
#!/bin/sh

# Install the goofys dependency for EMR Notebooks magics
sudo wget https://github.com/kahing/goofys/releases/latest/download/goofys -P /usr/bin/
sudo chmod ugo+x /usr/bin/goofys
```

Installieren Sie EMR Notebooks magicS

Note

Mit den Amazon EMR-Versionen 6.0 bis 6.9.0 und 5.0 bis 5.36.0 werden nur die `emr-notebooks-magics` Paketversionen 0.2.0 und höher unterstützt `%mount_workspace_dir` magic.

Gehen Sie wie folgt vor, um EMR Notebooks zu installieren magic-Geräte.

1. Führen Sie in Ihrem Notebook die folgenden Befehle aus, um das [emr-notebooks-magics](#)-Paket zu installieren.

```
%pip install boto3 --upgrade
%pip install botocore --upgrade
%pip install emr-notebooks-magics --upgrade
```

2. Starten Sie Ihren Kernel neu, um die EMR Notebooks zu laden magic-Geräte.
3. Überprüfen Sie Ihre Installation mit dem folgenden Befehl, der den Ausgabebeihfetext für `%mount_workspace_dir` anzeigen sollte.

```
%mount_workspace_dir?
```

Ein Workspace-Verzeichnis mit `%mount_workspace_dir` mounten

Die `%mount_workspace_dir` magic ermöglicht es Ihnen, Ihr Workspace-Verzeichnis auf Ihrem EMR-Cluster zu mounten, sodass Sie andere in Ihrem Verzeichnis gespeicherte Dateien, Module oder Pakete importieren und ausführen können.

Im folgenden Beispiel wird das gesamte Workspace-Verzeichnis auf einem Cluster bereitgestellt und das optionale `<--fuse-type>`-Argument angegeben, Goofys für das Mounten des Verzeichnisses zu verwenden.

```
%mount_workspace_dir . <--fuse-type goofys>
```

Um zu überprüfen, ob Ihr Workspace-Verzeichnis eingehängt ist, verwenden Sie das folgende Beispiel, um das aktuelle Arbeitsverzeichnis mit dem `ls`-Befehl anzuzeigen. Die Ausgabe sollte alle Dateien in Ihrem Workspace anzeigen.

```
%%sh  
ls
```

Wenn Sie mit den Änderungen in Ihrem Workspace fertig sind, können Sie das Workspace-Verzeichnis mit dem folgenden Befehl unmounten:

 Note

Ihr Workspace-Verzeichnis bleibt in Ihrem Cluster eingebunden, auch wenn der Workspace gestoppt oder getrennt wird. Sie müssen Ihr Workspace-Verzeichnis explizit unmounten.

```
%umount_workspace_dir
```

Herunterladen eines Amazon-S3-Objekts mit **%generate_s3_download_url**

Der `generate_s3_download_url`-Befehl erstellt eine vorsignierte URL für ein in Amazon S3 gespeichertes Objekt. Sie können die vorsignierte URL verwenden, um das Objekt auf Ihren lokalen Computer herunterzuladen. Sie könnten beispielsweise `generate_s3_download_url` ausführen, um das Ergebnis einer SQL-Abfrage herunterzuladen, die Ihr Code in Amazon S3 schreibt.

Die vorsignierte URL ist standardmäßig 60 Minuten lang gültig. Sie können die Ablaufzeit ändern, indem Sie eine Anzahl von Sekunden für das `--expires-in`-Kennzeichen angeben. `--expires-in 1800` erstellt beispielsweise eine URL, die 30 Minuten gültig ist.

Das folgende Beispiel generiert einen Download-Link für ein Objekt, indem der vollständige Amazon-S3-Pfad angegeben wird: ***s3://EXAMPLE-DOC-BUCKET/path/to/my/object***.

```
%generate_s3_download_url s3://EXAMPLE-DOC-BUCKET/path/to/my/object
```

Um mehr über die Verwendung von `generate_s3_download_url` zu erfahren, führen Sie den folgenden Befehl aus, um den Hilfetext anzuzeigen.

```
%generate_s3_download_url?
```

Führen Sie ein Notebook im Headless-Modus mit **%execute_notebook**

Mit `%execute_notebook` magic, können Sie ein anderes Notebook im Headless-Modus ausführen und sich die Ausgabe für jede Zelle ansehen, die Sie ausgeführt haben. Dieser magic erfordert

zusätzliche Berechtigungen für die Instance-Rolle, die Amazon EMR und Amazon EC2 gemeinsam nutzen. Führen Sie den `%execute_notebook?`-Befehl aus, um weitere Informationen zur Gewährung zusätzlicher Berechtigungen zu erhalten.

Während eines Auftrags mit langer Laufzeit wechselt Ihr System möglicherweise aufgrund von Inaktivität in den Standbymodus oder verliert vorübergehend die Internetverbindung. Dadurch könnte die Verbindung zwischen Ihrem Browser und dem Jupyter Server unterbrochen werden. In diesem Fall verlieren Sie möglicherweise die Ausgabe der Zellen, die Sie vom Jupyter Server ausgeführt und gesendet haben.

Wenn Sie das Notebook im Headless-Modus mit ausführen `%execute_notebook magic`, EMR Notebooks erfasst die Leistung der Zellen, die in Betrieb sind, auch wenn das lokale Netzwerk unterbrochen wird. EMR Notebooks speichert die Ausgabe inkrementell in einem neuen Notebook mit demselben Namen wie das Notebook, das Sie ausgeführt haben. EMR Notebooks platziert das Notebook dann in einem neuen Ordner innerhalb des Workspace. Headless-Läufe finden auf demselben Cluster statt und verwenden die Servicerolle `EMR_Notebook_DefaultRole`, aber zusätzliche Argumente können die Standardwerte ändern.

Verwenden Sie den folgenden Befehl, um ein Notebook im Headless-Modus auszuführen:

```
%execute_notebook <relative-file-path>
```

Verwenden Sie den folgenden Befehl, um eine Cluster-ID und eine Servicerolle für einen Headless-Lauf anzugeben:

```
%execute_notebook <notebook_name>.ipynb --cluster-id <emr-cluster-id> --service-role  
<emr-notebook-service-role>
```

Wenn Amazon EMR und Amazon eine Instance-Rolle EC2 gemeinsam nutzen, erfordert die Rolle die folgenden zusätzlichen Berechtigungen:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "elasticmapreduce:StartNotebookExecution",  
        "elasticmapreduce:DescribeNotebookExecution",  
        "ec2:DescribeInstances"  
      ]  
    }  
  ]  
}
```

```

    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::<AccountId>:role/EMR_Notebooks_DefaultRole"
  }
]
}

```

Note

Um `%execute_notebook` zu verwenden magic, installieren Sie das `emr-notebooks-magics` Paket, Version 0.2.3 oder höher.

Verwenden Sie mehrsprachige Notebooks mit Spark-Kernen

Jeder Jupyter-Notebook-Kernel hat eine Standardsprache. Die Standardsprache des Spark-Kernels ist beispielsweise Scala, und die Standardsprache des PySpark Kernels ist Python. Mit Amazon EMR 6.4.0 und höher unterstützt EMR Studio mehrsprachige Notizbücher. Das bedeutet, dass jeder Kernel in EMR Studio zusätzlich zur Standardsprache die folgenden Sprachen unterstützen kann: Python, Spark, R und Spark SQL.

Um diese Funktion zu aktivieren, geben Sie eine der folgenden Optionen an magic Befehle am Anfang einer beliebigen Zelle.

Sprache	Befehl
Python	<code>%%pyspark</code>
Scala	<code>%%scalaspark</code>
R	<code>%%rspark</code>
	Wird nicht für interaktive Workloads mit EMR Serverless unterstützt.

Sprache	Befehl
Spark-SQL	<code>%%sql</code>

Wenn diese Befehle aufgerufen werden, führen sie die gesamte Zelle innerhalb derselben Spark-Sitzung mit dem Interpreter der entsprechenden Sprache aus.

Die `%%pyspark` Zelle magic ermöglicht es Benutzern, PySpark Code in alle Spark-Kernel zu schreiben.

```
%%pyspark
a = 1
```

Die Zelle `%%sql` magic ermöglicht Benutzern die Ausführung von Spark-SQL-Code in allen Spark-Kernen.

```
%%sql
SHOW TABLES
```

Die `%%rspark` Zelle magic ermöglicht Benutzern die Ausführung von SparkR-Code in allen Spark-Kernen.

```
%%rspark
a <- 1
```

Die Zelle `%%scalaspark` magic ermöglicht Benutzern die Ausführung von Spark-Scala-Code in allen Spark-Kernen.

```
%%scalaspark
val a = 1
```

Teilen Sie Daten mit temporären Tabellen zwischen Sprachinterpretern

Mithilfe temporärer Tabellen können Sie Daten auch zwischen Sprachinterpretern austauschen. Das folgende Beispiel verwendet `%%pyspark` in einer Zelle, um eine temporäre Tabelle in Python zu erstellen, und verwendet `%%scalaspark` in der folgenden Zelle, um Daten aus dieser Tabelle in Scala zu lesen.

```
%%pyspark
df=spark.sql("SELECT * from nyc_top_trips_report LIMIT 20")
# create a temporary table called nyc_top_trips_report_view in python
df.createOrReplaceTempView("nyc_top_trips_report_view")
```

```
%%scalaspark
// read the temp table in scala
val df=spark.sql("SELECT * from nyc_top_trips_report_view")
df.show(5)
```

Überblick über Amazon EMR Notebooks

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Sie können Amazon EMR Notebooks zusammen mit Amazon EMR-Clustern verwenden, auf denen [Apache Spark](#) ausgeführt wird, um [Jupyter](#) Notebook und JupyterLab Schnittstellen innerhalb der Amazon EMR-Konsole zu erstellen und zu öffnen. Ein EMR-Notebook ist ein „Serverless“-Notebook, mit dem Sie Abfragen und Code ausführen können. Im Gegensatz zu einem herkömmlichen Notebook werden die Inhalte eines EMR-Notebooks – die Gleichungen, Abfragen, Modelle, der Code und der erläuternde Text in Notizbuchzellen – in einem Client ausgeführt. Die Befehle werden auf einem Kernel in dem EMR-Cluster ausgeführt. Notebook-Inhalte werden auch getrennt von den Cluster-Daten in Amazon S3 gespeichert, um eine sichere Speicherung und flexible Wiederverwendung zu gewährleisten.

Sie können einen Cluster starten, ein EMR Notebook zur Analyse anfügen und dann den Cluster beenden. Sie können auch ein Notebook schließen, das an einen ausgeführten Cluster angefügt ist, und zu einem anderen Cluster wechseln. Mehrere Benutzer können gleichzeitig Notebooks an denselben Cluster anfügen und in Amazon S3 Notebook-Dateien miteinander teilen. Diese Funktionen ermöglichen Ihnen die On-Demand-Ausführung von Clustern, um Kosten zu sparen und den Zeitaufwand für die Neukonfiguration von Notebooks für verschiedene Cluster und Datensätze zu reduzieren.

Sie können ein EMR-Notebook auch programmgesteuert mithilfe der Amazon-EMR-API ausführen, ohne mit der Amazon-EMR-Konsole interagieren zu müssen („Headless Execution“). Sie müssen eine Zelle in das EMR-Notebook aufnehmen, die über ein Parameter-Tag verfügt. Diese Zelle ermöglicht es einem Skript, neue Eingabewerte an das Notizbuch zu übergeben. Parametrisierte Notizbücher können mit unterschiedlichen Eingabewerten wiederverwendet werden. Es ist nicht erforderlich, Kopien desselben Notebooks zu erstellen, um es mit neuen Eingabewerten zu bearbeiten und auszuführen. Amazon EMR erstellt und speichert das Ausgabe-Notebook auf S3 für

jeden Lauf des parametrisierten Notebooks. API-Codebeispiele für EMR-Notebooks finden Sie unter [Beispiele für programmatische Befehle für EMR Notebooks](#).

Important

Die EMR-Notebooks-Funktion unterstützt Cluster, die Amazon-EMR-Versionen 5.18.0 und höher verwenden. Wir empfehlen, EMR Notebooks mit Clustern zu verwenden, die die neueste Version von Amazon EMR oder mindestens 5.30.0, 5.32.0 oder 6.2.0 verwenden. Mit diesen Versionen werden Jupyter-Kernel auf dem angefügten Cluster und nicht auf einer Jupyter-Instance ausgeführt werden. Dies verbessert die Leistung und erweitert Ihre Möglichkeiten von Kernen und Bibliotheken zu verbessern. Weitere Informationen finden Sie unter [Unterschiede in den Funktionalitäten nach Cluster-Release-Version](#).

Es fallen Gebühren für Amazon-S3-Speicher und für Amazon-EMR-Cluster an.

Amazon EMR Notebooks sind als Amazon EMR Studio Workspaces in der Konsole verfügbar.

Der Übergang von EMR Notebooks zu Workspaces

In der [neuen Amazon-EMR-Konsole](#) haben wir EMR Notebooks mit Amazon EMR Studio Workspaces zu einem einzigen Erlebnis zusammengeführt. Wenn Sie ein EMR Studio verwenden, können Sie verschiedene Workspaces erstellen und konfigurieren, um Notebooks zu organisieren und auszuführen. Wenn Sie Amazon EMR-Notebooks in der alten Konsole hatten, sind sie in der Konsole als EMR Studio Workspaces verfügbar.

Amazon EMR hat diese neuen EMR Studio Workspaces für Sie erstellt. Die Anzahl der Studios, die wir erstellt haben, entspricht der Anzahl der Distinct, VPCs die Sie von EMR Notebooks verwenden. Wenn Sie beispielsweise eine Verbindung zu EMR-Clustern herstellen, die sich VPCs von EMR Notebooks unterscheiden, haben wir zwei neue EMR-Studios erstellt. Ihre Notebooks werden auf die neuen Studios verteilt.

 Important

Wir haben die Option zum Erstellen neuer Notizbücher in der alten Amazon-EMR-Konsole deaktiviert. Verwenden Sie stattdessen Workspace erstellen in der neuen Amazon-EMR-Konsole.

Weitere Informationen zu Amazon EMR Studio WorkSpaces finden Sie unter [Lernen Sie EMR Studio-Arbeitsbereiche kennen](#). Eine konzeptionelle Übersicht über EMR Studio finden Sie unter [Workspaces](#) auf der Seite [Wie Amazon EMR Studio funktioniert](#).

Was müssen Sie als Nächstes tun?

Sie können Ihre vorhandenen Notizbücher zwar weiterhin in der alten Konsole verwenden, wir empfehlen jedoch, stattdessen Amazon EMR Studio Workspaces in der Konsole zu verwenden. Sie müssen zusätzliche Rollenberechtigungen konfigurieren, um die [Funktionen in EMR Studio zu aktivieren, die in EMR-Notebooks nicht verfügbar sind](#).

 Note

Um bestehende EMR-Notebooks als EMR Studio Workspaces anzuzeigen und neue Workspaces zu erstellen, müssen Benutzer mindestens über `elasticmapreduce:ListStudios`- und `elasticmapreduce:CreateStudioPresignedUrl`-Berechtigungen für ihre Rollen verfügen. Um auf alle Features von EMR Studio zuzugreifen, finden Sie unter [Aktivierung der Feature von EMR Studio für Benutzer von EMR-Notebooks](#) eine vollständige Liste der zusätzlichen Berechtigungen, die Benutzer von EMR-Notebooks benötigen.

Verbesserte Funktionen in EMR Studio, die über EMR-Notebooks hinausgehen

Mit Amazon EMR Studio können Sie die folgenden Funktionen einrichten und verwenden, die mit EMR-Notebooks nicht verfügbar sind:

- [Durchsuchen und Verbinden mit EMR-Clustern aus Jupyterlab heraus](#)
- [Durchsuchen und Anhängen an virtuelle EMR-Notebook-Cluster aus Jupyterlab heraus](#)

- [Verbindung zu Git-Repositories aus Jupyterlab heraus](#)
- [Zusammenarbeit mit anderen Mitgliedern Ihres Teams beim Schreiben und Ausführen von Notebook-Code](#)
- [Durchsuchen von Daten mit SQL Explorer](#)
- [Bereitstellung von EMR-Clustern mit Service Catalog](#)

Eine vollständige Liste der Funktionen von Amazon EMR Studio finden Sie unter [Hauptfeatures von EMR Studio](#).

Aktivierung der Feature von EMR Studio für Benutzer von EMR-Notebooks

Die neuen EMR Studios, die wir im Rahmen dieser Zusammenführung erstellen werden, verwenden die bestehende `EMR_Notebooks_DefaultRole`-IAM-Rolle als EMR-Studio-Service-Rolle.

Benutzer, die von EMR-Notebooks zu EMR Studio wechseln und die zusätzlichen Funktionen von EMR Studio nutzen möchten, benötigen mehrere neue Rollenberechtigungen. Fügen Sie den Rollen Ihrer EMR-Notebooks-Benutzer, die EMR Studio verwenden möchten, die folgenden Berechtigungen hinzu.

Note

Um bestehende EMR-Notebooks als EMR Studio Workspaces anzuzeigen und neue Workspaces zu erstellen, müssen Benutzer mindestens über `elasticmapreduce:ListStudios`- und `elasticmapreduce:CreateStudioPresignedUrl`-Berechtigungen für ihre Rollen verfügen. Um alle Features von EMR Studio zu verwenden, fügen Sie alle unten aufgeführten Berechtigungen hinzu. Admin-Benutzer benötigen außerdem die Erlaubnis, ein EMR Studio zu erstellen und zu verwalten. Weitere Informationen finden Sie unter [Administratorberechtigungen zum Erstellen und Verwalten eines EMR Studios](#).

```
"elasticmapreduce:DescribeStudio",  
"elasticmapreduce:ListStudios",  
"elasticmapreduce:CreateStudioPresignedUrl",  
"elasticmapreduce:UpdateEditor",  
"elasticmapreduce:PutWorkspaceAccess",  
"elasticmapreduce>DeleteWorkspaceAccess",
```



```
"elasticmapreduce:ListWorkspaceAccessIdentities",  
"emr-containers:ListVirtualClusters",  
"emr-containers:DescribeVirtualCluster",  
"emr-containers:ListManagedEndpoints",  
"emr-containers:DescribeManagedEndpoint",  
"emr-containers:CreateAccessTokenForManagedEndpoint",  
"emr-containers:ListJobRuns",  
"emr-containers:DescribeJobRun",  
"servicecatalog:SearchProducts",  
"servicecatalog:DescribeProduct",  
"servicecatalog:DescribeProductView",  
"servicecatalog:DescribeProvisioningParameters",  
"servicecatalog:ProvisionProduct",  
"servicecatalog:UpdateProvisionedProduct",  
"servicecatalog:ListProvisioningArtifacts",  
"servicecatalog:DescribeRecord",  
"servicecatalog:ListLaunchPaths",  
"cloudformation:DescribeStackResources"
```

Die folgenden Berechtigungen sind ebenfalls erforderlich, um die Funktionen für die Zusammenarbeit in EMR Studio zu nutzen, waren jedoch für EMR-Notebooks nicht erforderlich.

```
"sso-directory:SearchUsers",  
"iam:GetUser",  
"iam:GetRole",  
"iam:ListUsers",  
"iam:ListRoles",  
"sso:GetManagedApplicationInstance"
```

Anforderungen, Unterschiede in den Release-Versionen und Sicherheit für EMR Notebooks

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Beachten Sie die folgenden Anforderungen, Unterschiede in den Release-Versionen, Sicherheitsinformationen und andere Überlegungen, wenn Sie Cluster erstellen und Lösungen mit EMR Notebook entwickeln.

Cluster-Voraussetzungen

- Amazon EMR Block Public Access aktivieren – Durch den eingehenden Zugriff auf einen Cluster können Cluster-Benutzer Notebook-Kernel ausführen. Stellen Sie sicher, dass nur autorisierte Benutzer auf den Cluster zugreifen können. Es wird dringend empfohlen, den öffentlichen Zugriff zu blockieren und eingehenden SSH-Datenverkehr auf vertrauenswürdige Quellen zu beschränken. Weitere Informationen erhalten Sie unter [Verwenden von Amazon EMR Block Public Access](#) und [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).
- Kompatiblen Cluster verwenden – Ein Cluster, der an ein Notebook angefügt ist, muss die folgenden Voraussetzungen erfüllen:
 - Es werden nur Cluster, die mit Amazon EMR erstellt wurden, unterstützt. Sie können innerhalb von Amazon EMR unabhängig einen Cluster erstellen und dann ein EMR-Notebook anfügen. Sie können einen kompatiblen Cluster auch erstellen, wenn Sie ein EMR-Notebook erstellen.
 - Es werden nur Cluster, die mit Amazon EMR Version 5.18.0 oder höher erstellt wurden, unterstützt. Siehe [the section called “Unterschiede in den Funktionalitäten nach Cluster-Release-Version”](#).
 - Cluster, die mithilfe von EC2 Amazon-Instances mit AMD EPYC-Prozessoren erstellt wurden — zum Beispiel die Instance-Typen m5a.* und r5a.* — werden nicht unterstützt.
 - EMR-Notebooks funktioniert nur mit Clustern, die mit der VisibleToAllUsers-Einstellung auf true erstellt wurden. VisibleToAllUsers ist standardmäßig true.
 - Der Cluster muss innerhalb einer EC2 VPC gestartet werden. Öffentliche und private Subnetze werden unterstützt. Die EC2 -Classic-Plattform wird nicht unterstützt.
 - Hadoop, Spark und Livy müssen auf dem Cluster installiert sein. Andere Anwendungen können installiert werden, aber EMR Notebook unterstützt derzeit nur Spark-Cluster.

Important

Für Amazon-EMR-Versionen 5.32.0 und höher oder 6.2.0 und höher muss auf Ihrem Cluster auch die Jupyter Enterprise Gateway-Anwendung ausgeführt werden, um mit EMR-Notebooks zu funktionieren.

- Cluster mit Kerberos-Authentifizierung werden nicht unterstützt.

- Mit integrierte Cluster AWS Lake Formation unterstützen nur die Installation von Bibliotheken für Notebooks. Die Installation von Kernen und Bibliotheken auf dem Cluster wird nicht unterstützt.
- Cluster mit mehreren Primärknoten werden nicht unterstützt.
- Cluster, die EC2 Amazon-Instances verwenden, die auf AWS Graviton2 basieren, werden nicht unterstützt.

Unterschiede in den Funktionalitäten nach Cluster-Release-Version

Wir empfehlen dringend, EMR-Notebooks mit Clustern zu verwenden, die mit den Amazon-EMR-Versionen 5.30.0, 5.32.0 oder höher oder 6.2.0 oder höher erstellt wurden. Mit diesen Versionen führt EMR Notebooks-Kernel auf dem angeschlossenen Amazon-EMR-Cluster aus. Kernel und Bibliotheken können direkt auf dem Cluster-Primärknoten installiert werden. Die Verwendung von EMR-Notebooks mit diesen Cluster-Versionen hat folgende Vorteile:

- Verbesserte Leistung — Notebook-Kernel werden auf Clustern mit von Ihnen ausgewählten EC2 Instance-Typen ausgeführt. Frühere Versionen führen Kernel auf einer spezialisierten Instance aus, die nicht in der Größe geändert, auf die nicht zugegriffen und die nicht angepasst werden kann.
- Möglichkeit zum Hinzufügen und Anpassen von Kernen – Sie können eine Verbindung zum Cluster herstellen, um Kernel-Pakete mit conda und pip zu installieren. Darüber hinaus wird die pip-Installation mithilfe von Terminal-Befehlen innerhalb von Notebook-Zellen unterstützt. In früheren Versionen waren nur vorinstallierte Kernel verfügbar (Python PySpark, Spark und SparkR). Weitere Informationen finden Sie unter [Installieren von Kernels und Python-Bibliotheken auf einem Cluster-Primärknoten](#).
- Möglichkeit, Python-Bibliotheken zu installieren – Sie können [Python-Bibliotheken mit conda und pip auf dem Cluster-Primärknoten](#) installieren. Wir empfehlen die Verwendung von conda. In früheren Versionen wurden nur Bibliotheken für [Notebooks unterstützt](#). PySpark

Unterstützte EMR-Notebooks-Features nach Cluster-Version

Cluster-Version	Bibliotheken für Notebooks PySpark	Kernel-Installation auf dem Cluster	Installation der Python-Bibliothek auf Primärknoten
Früher als 5.18.0	EMR Notebooks werden nicht unterstützt		
5.18.0–5.25.0	Nein	Nein	Nein

Cluster-Version	Bibliotheken für Notebooks PySpark	Kernel-Installation auf dem Cluster	Installation der Python-Bibliothek auf Primärknoten
5.26.0–5-29.0	Ja	Nein	Nein
5.30.0	Ja	Ja	Ja
6.0.0	Nein	Nein	Nein
5.32.0 und höher und 6.2.0 und höher	Ja	Ja	Ja

Limits für gleichzeitig angefügte EMR-Notebooks

Wenn Sie einen Cluster erstellen, der Notebooks unterstützt, sollten Sie den EC2 Instanztyp des primären Clusterknotens berücksichtigen. Die Speicherbeschränkungen dieser EC2 Instanz bestimmen die Anzahl der Notebooks, die gleichzeitig bereit sein können, Code und Abfragen auf dem Cluster auszuführen.

EC2 Instanztyp des primären Knotens	Anzahl der EMR Notebooks
*.medium	2
*.large	4
*.xlarge	8
*.2xlarge	16
*.4xlarge	24
*.8xlarge	24
*.16xlarge	24

Jupyter Notebook und Python-Versionen

EMR-Notebooks führt [Jupyter Notebook Version 6.0.2](#) und Python 3.6.5 aus, unabhängig von der Amazon-EMR-Version des angefügten Clusters.

Sicherheitsüberlegungen

Verwenden verschlüsselter S3-Standorte

Wenn Sie einen verschlüsselten Speicherort in Amazon S3 zum Speichern von Notebook-Dateien angeben, müssen Sie die [Servicerolle für EMR Notebooks](#) als Schlüsselbenutzer einrichten. Die Standard-Servicerolle ist `EMR_Notebooks_DefaultRole`. Wenn Sie einen AWS KMS Schlüssel für die Verschlüsselung verwenden, finden Sie weitere Informationen [unter Verwenden von Schlüsselrichtlinien in AWS KMS](#) im AWS Key Management Service Entwicklerhandbuch und im [Support-Artikel zum Hinzufügen von Schlüsselbenutzern](#).

Verwendung von Cookies mit Hosting-Domains

Um die Sicherheit der Anwendungen außerhalb der Konsole zu erhöhen, die Sie möglicherweise mit Amazon EMR verwenden, sind die Anwendungs-Hosting-Domains in der Public Suffix List (PSL) registriert. Zu diesen Hosting-Domains gehören beispielsweise die folgenden: `emrstudio-prod.us-east-1.amazonaws.com`, `emrnotebooks-prod.us-east-1.amazonaws.com`, `emrappui-prod.us-east-1.amazonaws.com`. Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host-`-Präfix zu verwenden, falls Sie jemals sensible Cookies im Standard-Domainnamen einrichten müssen. Diese Vorgehensweise hilft Ihnen dabei, Ihre Domain vor CSRF (Cross-Site Request Forgery Attempts, Anforderungsfälschung zwischen Websites)-Versuchen zu schützen. Weitere Informationen finden Sie hier: [Set-Cookie](#)Seite im Mozilla Developer Network.

Erstellen Sie ein Notizbuch in EMR Studio

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Sie erstellen ein EMR-Notebook mit der alten Amazon-EMR-Konsole. Das Erstellen von Notizbüchern mit der AWS CLI oder der Amazon EMR-API wird nicht unterstützt.

Erstellen eines EMR-Notebooks

1. Öffnen Sie die Amazon-EMR-Konsole unter <https://console.aws.amazon.com/elasticmapreduce/>.
2. Wählen Sie Notebooks, Create notebook (Notebook erstellen).
3. Geben Sie einen Notebook name (Notebook-Namen) und optional eine Notebook description (Notebook-Beschreibung) ein.
4. Wenn Sie einen aktiven Cluster haben, den Sie an das Notebook anfügen möchten, behalten Sie die Standardeinstellung Vorhandenen Cluster auswählen bei und wählen Wählen aus. Wählen Sie anschließend einen Cluster aus der Liste und Cluster wählen aus. Informationen zu den Clusteranforderungen für EMR Notebooks finden Sie unter [Anforderungen, Unterschiede in den Release-Versionen und Sicherheit für EMR Notebooks](#).

—oder—

Wählen Sie Create a cluster (Cluster erstellen), geben Sie einen Clusternamen ein und wählen Sie Optionen gemäß den folgenden Richtlinien aus. Der Cluster wird unter Verwendung von On-Demand-Instances in der Standard-VPC für das Konto erstellt.

Einstellung	Beschreibung
Cluster name	Der Anzeigename, der zum Identifizieren des Clusters verwendet wird.
Veröffentlichung	Kann nicht geändert werden. Standardmäßig wird die neueste Amazon EMR-Release-Version (5.36.2) verwendet.
Anwendungen	Kann nicht geändert werden. Listet die Anwendungen auf, die auf dem Cluster installiert sind.
Instance	Geben Sie die Anzahl der Instances ein und wählen Sie den Instance-Typ aus. EC2 Eine Instance wird für den Primärknoten verwendet. Der Rest wird für Core-Knoten

Einstellung	Beschreibung
	verwendet. Der Instance-Typ bestimmt die Anzahl der Notebooks, die gleichzeitig an den Cluster angefügt sein können. Weitere Informationen finden Sie unter Limits für gleichzeitig angefügte EMR-Notebooks.
EMR role (EMR-Rolle)	Behalten Sie die Standardeinstellung bei, oder wählen Sie den Link aus, um eine benutzerdefinierte Servicerolle für Amazon EMR anzugeben. Weitere Informationen finden Sie unter Servicerolle für Amazon EMR (EMR-Rolle).
EC2 Instance-Profil	Behalten Sie die Standardeinstellung bei oder wählen Sie den Link, um eine benutzerdefinierte Servicerolle für EC2 Instanzen anzugeben. Weitere Informationen finden Sie unter Servicerolle für EC2 Cluster-Instances (EC2Instance-Profil).
EC2 Schlüsselpaar	Wählen Sie ein EC2 key pair, um eine Verbindung zu Cluster-Instances herstellen zu können. Weitere Informationen finden Sie unter Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her.

Einstellung	Beschreibung
Automatische Beendigung	Automatische Beendigung wird für Amazon-EMR-Versionen 5.30.0 und 6.1.0 und höher unterstützt. Aktivieren Sie das Kontrollkästchen, um die automatische Beendigung zu aktivieren, und geben Sie dann die Leerlaufzeit an, nach der der Cluster automatisch heruntergefahren werden soll. Weitere Informationen finden Sie unter Verwenden einer automatischen Terminierungsrichtlinie für die Amazon EMR-Clusterbereinigung.

- Wählen Sie unter Security groups (Sicherheitsgruppen) die Option Use default security groups (Standardsicherheitsgruppen verwenden). Alternativ können Sie Choose security groups (Sicherheitsgruppen auswählen) wählen und dann benutzerdefinierte Sicherheitsgruppen auswählen, die in der VPC des Clusters verfügbar sind. Sie wählen eine Sicherheitsgruppe für die primäre Instance und eine andere für den Notebook-Service aus. Weitere Informationen finden Sie unter [the section called "Sicherheitsgruppe für EMR Notebooks"](#).
- Behalten Sie für AWS -Servicerolle die Standardeinstellung bei oder wählen Sie eine benutzerdefinierte Rolle aus der Liste aus. Die Client-Instance für das Notebook verwendet diese Rolle. Weitere Informationen finden Sie unter [Servicerolle für EMR Notebooks](#).
- Wählen Sie unter Notebook-Speicherort den Speicherort für die Notebook-Datei in Amazon S3 aus oder geben Sie einen eigenen Speicherort an. Wenn der Bucket und Ordner nicht vorhanden sind, werden sie von Amazon EMR erstellt.

Amazon EMR erstellt einen Ordner mit der Notebook-ID als Ordnernamen und speichert das Notebook in einer Datei mit dem Namen *NotebookName*.ipynb. Wenn Sie zum Beispiel den Amazon-S3-Speicherort `s3://amzn-s3-demo-bucket/MyNotebooks` für ein Notebook mit dem Namen `MyFirstEMRManagedNotebook` angeben, wird die Notebook-Datei unter `s3://amzn-s3-demo-bucket/MyNotebooks/NotebookID/MyFirstEMRManagedNotebook.ipynb` gespeichert.

Wenn Sie einen verschlüsselten Speicherort in Amazon S3 angeben, müssen Sie [Servicerolle für EMR Notebooks](#) als Schlüsselbenutzer einrichten. Die Standard-Servicerolle ist

EMR_Notebooks_DefaultRole. Wenn Sie einen AWS KMS Schlüssel für die Verschlüsselung verwenden, finden Sie weitere Informationen [unter Verwenden von Schlüsselrichtlinien in AWS KMS](#) im AWS Key Management Service Entwicklerhandbuch und im [Support-Artikel zum Hinzufügen von Schlüsselbenutzern](#).

8. (Optional) Wenn Sie ein Git-basiertes Repository zu Amazon EMR hinzugefügt haben, das Sie mit diesem Notebook verknüpfen möchten, wählen Sie Git-Repository, wählen Sie auf Repository auswählen und wählen Sie dann ein Repository aus der Liste aus. Weitere Informationen finden Sie unter [Verknüpfen von Git-basierten Repositories mit EMR Notebooks](#).
9. Wählen Sie optional Tags und fügen Sie dann alle zusätzlichen Schlüssel-Wert-Tags für das Notebook hinzu.

 Important

Für den Zugriff wird ein Standard-Tag verwendet, bei dem die Schlüsselzeichenfolge auf `creatorUserID` und der Wert auf Ihre IAM-Benutzer-ID festgelegt ist. Wir empfehlen, dieses Tag nicht zu ändern oder zu entfernen, da es für die Zugriffssteuerung verwendet werden kann. Weitere Informationen finden Sie unter [Verwenden Sie Cluster- und Notebook-Tags mit IAM-Richtlinien für die Zugriffskontrolle](#).

10. Klicken Sie auf Create Notebook (Notebook erstellen).

Arbeiten mit EMR-Notebooks

 Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Nach dem Erstellen eines EMR-Notebooks dauert es einige Zeit, bis das Notebook gestartet wird. Der Status in der Liste Notebooks zeigt Starting (Wird gestartet) an. Sie können ein Notebook öffnen, wenn der Status Ready (Bereit) lautet. Es kann etwas länger dauern, bis ein Notebook den Status Ready (Bereit) anzeigt, wenn Sie einen Cluster mit diesem zusammen erstellt haben.

 Tip

Aktualisieren Sie Ihren Browser oder wählen Sie das Aktualisierungssymbol über der Liste „Notebooks“, um den Notebookstatus zu aktualisieren.

Grundlegendes zum Notebook-Status

Ein EMR-Notebook kann unter Status in der Liste Notebooks die folgenden Optionen anzeigen.

Status	Bedeutung
Bereit	Sie können das Notebook mithilfe des Notebook-Editors öffnen. Wenn ein Notebook den Status Ready (Bereit) aufweist, können Sie es anhalten oder löschen. Um Cluster zu wechseln, müssen Sie das Notebook zuerst anhalten. Wenn ein Notebook mit dem Status Ready (Bereit) für einen langen Zeitraum inaktiv ist, wird es automatisch angehalten.
Wird gestartet	Das Notebook wird erstellt und an den Cluster angehängt. Während ein Notebook gestartet wird, können Sie den Notebook-Editor nicht öffnen, anhalten oder löschen und Cluster nicht wechseln.
Ausstehend	Das Notebook wurde erstellt und wartet darauf, dass die Integration mit dem Cluster abgeschlossen ist. Der Cluster stellt möglicherweise weiterhin Ressourcen bereit oder reagiert auf andere Anfragen. Sie können den Notebook-Editor mit dem Notebook im lokalen Modus öffnen. Code, der von Cluster-Prozessen abhängt, wird nicht ausgeführt und schlägt fehl.
Wird angehalten	Das Notebook wird heruntergefahren oder der Cluster, an den das Notebook angehängt ist,

Status	Bedeutung
	wird beendet. Während ein Notebook beendet wird, können Sie den Notebook-Editor nicht öffnen, anhalten oder löschen und Cluster nicht wechseln.
Angehalten	Das Notebook wurde heruntergefahren. Sie können das Notebook auf demselben Cluster starten, solange der Cluster noch ausgeführt wird. Sie können Cluster wechseln und den Cluster löschen.
Löschen	Der Cluster wird aus der Liste der verfügbaren Cluster entfernt. Die Notebook-Datei <i>NotebookName</i> .ipynb verbleibt in Amazon S3 und es fallen weiterhin Gebühren für die Speicherung an.

Arbeiten mit dem Notebook-Editor

Ein Vorteil der Verwendung eines EMR-Notebooks besteht darin, dass Sie das Notebook in Jupyter oder JupyterLab direkt von der Konsole aus starten können.

Bei EMR Notebooks ist der Notebook-Editor, auf den Sie von der Amazon EMR-Konsole aus zugreifen, der vertraute Open-Source-Jupyter Notebook-Editor oder JupyterLab. Da der Notebook-Editor innerhalb der Amazon-EMR-Konsole gestartet wird, ist es effizienter, den Zugriff hierüber zu konfigurieren als mit einem auf einem Amazon-EMR-Cluster gehosteten Notebook. Sie müssen den Client eines Benutzers nicht so konfigurieren, dass er über Webzugriff über SSH, Sicherheitsgruppenregeln und Proxy-Konfigurationen verfügt. Wenn ein Benutzer über ausreichende Berechtigungen verfügt, kann er den Notebook-Editor einfach innerhalb der Amazon-EMR-Konsole öffnen.

Ein EMR-Notebook kann immer nur von einem Benutzer aus mit Amazon EMR geöffnet sein. Wenn ein anderer Benutzer versucht, ein bereits geöffnetes EMR-Notebook zu öffnen, tritt ein Fehler auf.

Important

Amazon EMR erstellt eine eindeutige vorseignierte URL für jede Notebook-Editorsitzung, die nur für kurze Zeit gültig ist. Wir empfehlen, dass Sie die Notebook-Editor-URL nicht freigeben. Dies stellt ein Sicherheitsrisiko dar, da Empfänger der URL Ihre Berechtigungen zur Bearbeitung des Notebooks übernehmen und Notebook-Code für die Lebensdauer der URL ausführen. Wenn andere Benutzer Zugriff auf ein Notebook benötigen, legen Sie über Berechtigungsrichtlinien Berechtigungen für deren Benutzer fest und stellen Sie sicher, dass die Servicerolle für EMR-Notebooks Zugriff auf den Amazon-S3-Speicherort hat. Weitere Informationen erhalten Sie unter [the section called “Sicherheit”](#) und [Servicerolle für EMR Notebooks](#).

So öffnen Sie den Notebook-Editor für ein EMR-Notebook

1. Wählen Sie einen Notebook mit dem Status Ready (Bereit) oder Pending (Ausstehend) in der Liste Notebooks aus.
2. Wählen Sie „Öffnen in“ oder „In Jupyter öffnen“. JupyterLab

Für den JupyterLab oder den Jupyter Notebook-Editor wird ein neuer Browser-Tab geöffnet.

3. Wählen Sie im Menü Kernel die Option Change Kernel (Kernel ändern) und wählen Sie dann den Kernel für Ihre Programmiersprache aus.

Sie können jetzt Code innerhalb des Notebook-Editors schreiben und ausführen.

Speichern der Inhalte eines Notebooks

Wenn Sie im Notebook-Editor arbeiten, werden die Inhalte von Notebook-Zellen und Ausgaben automatisch regelmäßig in der Notebook-Datei in Amazon S3 gespeichert. Ein Notebook ohne Änderungen seit der letzten Bearbeitung von Zellen zeigt den Eintrag (autosaved) (automatisch gespeichert) neben dem Notebook-Namen im Editor an. Wenn Änderungen noch nicht gespeichert wurden, wird unsaved changes (nicht gespeicherte Änderungen) angezeigt.

Sie können ein Notebook manuell speichern. Wählen Sie im Menü Datei die Option Speichern und Checkpoint oder drücken Sie STRG+S. Dadurch wird eine Datei mit dem Namen *NotebookName*.ipynb in einem Checkpoints-Ordner innerhalb des Notebookordners in Amazon S3 erstellt. Beispiel, s3://*amzn-s3-demo-bucket*/*MyNotebookFolder*/*NotebookID*/

checkpoints/*NotebookName*.ipynb. Nur die aktuelle Prüfpunktdatei wird an diesem Speicherort gespeichert.

Wechseln von Clustern

Sie können den Cluster wechseln, an den ein EMR-Notebook angehängt ist, ohne die Inhalte des Notebooks selbst zu ändern. Sie können Cluster nur für Notebooks mit dem Status Stopped (Angehalten) wechseln.

So ändern Sie den Cluster eines EMR-Notebooks

1. Wenn das Notebook, das Sie wechseln möchten, ausgeführt wird, wählen Sie dieses in der Liste Notebooks und anschließend Stop (Anhalten) aus.
2. Wenn das Notebook den Status Stopped (Angehalten) aufweist, wählen Sie das Notebook in der Liste Notebooks und anschließend View details (Details anzeigen) aus.
3. Wählen Sie Change cluster (Cluster wechseln).
4. Wenn Sie einen aktiven Cluster haben, auf dem Hadoop, Spark und Livy ausgeführt werden und an den Sie das Notebook anfügen möchten, behalten Sie die Standardeinstellung bei und wählen Sie einen Cluster aus der Liste aus. Es werden nur Cluster aufgeführt, die diesen Anforderungen entsprechen.

–oder–

Wählen Sie Create a cluster (Cluster erstellen) und anschließend die Clusteroptionen. Weitere Informationen finden Sie unter [Cluster-Voraussetzungen](#).

5. Wählen Sie eine Option für Security groups (Sicherheitsgruppen) und anschließend Change cluster and start notebook (Cluster wechseln und Notebook starten).

Löschen von Notebooks und Notebook-Dateien

Wenn Sie ein EMR-Notebook mithilfe der Amazon-EMR-Konsole löschen, löschen Sie das Notebook aus der Liste der verfügbaren Notebooks. Notebook-Dateien verbleiben jedoch in Amazon S3 und es fallen weiterhin Speicherkosten an.

So löschen Sie ein Notizbuch und entfernen die zugehörigen Dateien

1. Öffnen Sie die Amazon-EMR-Konsole unter <https://console.aws.amazon.com/elasticmapreduce/>.

2. Wählen Sie Notebooks, wählen Sie Ihr Notebook aus der Liste und anschließend View details (Details anzeigen) aus.
3. Wählen Sie das Ordnersymbol neben Notebook location (Notebook-Speicherort) und kopieren Sie die URL mit dem Muster `s3://MyNotebookLocationPath/NotebookID/`.
4. Wählen Sie Löschen aus.

Das Notebook wird aus der Liste entfernt und die Notebook-Details können nicht mehr angezeigt werden.

5. Befolgen Sie die Anweisungen für [Wie kann ich Ordner aus einem S3-Bucket löschen?](#) im Benutzerhandbuch für Amazon Simple Storage Service. Navigieren Sie zum Bucket und Ordner aus Schritt 3.

–oder–

Wenn Sie den AWS CLI installiert haben, öffnen Sie eine Befehlszeile und geben Sie den Befehl am Ende dieses Absatzes ein. Ersetzen Sie den Amazon-S3-Speicherort mit dem oben kopierten Speicherort. Stellen Sie sicher, dass der mit den Zugriffsschlüsseln eines Benutzers konfiguriert AWS CLI ist, der berechtigt ist, den Amazon S3 S3-Standort zu löschen. Weitere Informationen finden Sie unter [Konfigurieren der AWS CLI](#) im AWS Command Line Interface -Leitfaden.

```
aws s3 rm s3://MyNotebookLocationPath/NotebookID
```

Freigeben von Notebook-Dateien

Jedes EMR-Notebook wird in Amazon S3 als eine Datei mit dem Namen *NotebookName*.ipynb gespeichert. Solange eine Notebook-Datei mit derselben Version von Jupyter Notebook kompatibel ist, auf der EMR-Notebooks basiert, können Sie das Notizbuch als EMR-Notebook öffnen.

Der einfachste Weg, eine Notebook-Datei von einem anderen Benutzer zu öffnen, besteht darin, die *.ipynb-Datei eines anderen Benutzers in Ihrem lokalen Dateisystem zu speichern und dann die Upload-Funktion in Jupyter und Editoren zu verwenden. JupyterLab

Mithilfe dieses Verfahrens können Sie von anderen Personen freigegebene EMR-Notebooks und in der Jupyter-Community freigegebene Notebooks verwenden oder aus der Konsole gelöschte Notebooks wiederherstellen, sofern die Notebook-Datei noch vorhanden ist.

So verwenden Sie eine andere Notebook-Datei als Basis für ein EMR-Notebook

1. Bevor Sie fortfahren, schließen Sie den Notebook-Editor für alle Notebooks, mit denen Sie arbeiten werden, und halten Sie dann das Notebook an, wenn es sich um ein EMR-Notebook handelt.
2. Erstellen Sie ein EMR-Notebook und geben Sie einen Namen dafür ein. Der Name, den Sie für das Notebook eingeben, wird der Name der Datei sein, die Sie ersetzen müssen. Der neue Dateiname muss genau mit diesem Dateinamen übereinstimmen.
3. Notieren Sie sich den Speicherort in Amazon S3, den Sie für das Notebook wählen. Die Datei, die Sie ersetzen, befindet sich in einem Ordner mit einem Pfad und Dateinamen, die dem folgenden Muster entsprechen:
`s3://MyNotebookLocation/NotebookID/MyNotebookName.ipynb`.
4. Halten Sie das Notebook an.
5. Ersetzen Sie die alte Notebook-Datei im Amazon-S3-Speicherort mit der neuen Datei, die denselben Namen trägt.

Der folgende AWS CLI Befehl für Amazon S3 ersetzt eine Datei, die auf einem lokalen Computer gespeichert wurde, der `SharedNotebook.ipynb` für ein EMR-Notizbuch aufgerufen wird, durch den in Amazon S3 `amzn-s3-demo-bucket/MyNotebooksFolder` angegebenen Namen `MyNotebook`, eine ID von und die Dateie-`12A3BCDEFJHIJKLMNOP045PQRST`, mit der sie erstellt wurde. Weitere Informationen zur Verwendung der Amazon S3 Konsole zum Kopieren und Ersetzen von Dateien finden Sie unter [Objekte hochladen, herunterladen und verwalten](#) im Benutzerhandbuch für Amazon Simple Storage Service..

```
aws s3 cp SharedNotebook.ipynb s3://amzn-s3-demo-bucket/
MyNotebooksFolder/-12A3BCDEFJHIJKLMNOP045PQRST/MyNotebook.ipynb
```

Beispiele für programmatische Befehle für EMR Notebooks

Übersicht

Sie können EMR-Notebooks mit Ausführung APIs über ein Skript oder über die Befehlszeile ausführen. Wenn Sie EMR-Notebookausführungen außerhalb der AWS Konsole starten, beenden, auflisten und beschreiben, können Sie ein EMR-Notebook programmgesteuert steuern. Sie können verschiedene Parameterwerte an ein Notebook mit einer parametrisierten Notebookzelle übergeben.

Dadurch entfällt die Notwendigkeit, für jeden neuen Satz von Parameterwerten eine Kopie des Notebooks zu erstellen. Weitere Informationen finden Sie unter [API-Aktionen in Amazon EMR](#).

Sie können EMR-Notebook-Ausführungen mit CloudWatch Amazon-Ereignissen und stapeln. AWS Lambda Weitere Informationen finden Sie unter [Verwendung AWS Lambda mit Amazon CloudWatch Events](#).

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Rollenberechtigungen für die programmatische Ausführung

Um die programmgesteuerte Ausführung mit EMR Notebooks zu verwenden, müssen Sie Benutzerberechtigungen mit den folgenden Richtlinien konfigurieren:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowExecutionActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:StartNotebookExecution",
        "elasticmapreduce:DescribeNotebookExecution",
        "elasticmapreduce:ListNotebookExecutions"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowPassingServiceRole",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/EMR_Notebooks_DefaultRole"
```



```

    }
  ]
}

```

Wenn Sie EMR Notebooks programmgesteuert auf einem EMR-Notebooks-Cluster ausführen, müssen Sie die folgenden zusätzlichen Berechtigungen hinzufügen:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRetrievingManagedEndpointCredentials",
      "Effect": "Allow",
      "Action": [
        "emr-containers:GetManagedEndpointSessionCredentials"
      ],
      "Resource": [
        "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-cluster-id/endpoints/managed-endpoint-id"
      ],
      "Condition": {
        "StringEquals": {
          "emr-containers:ExecutionRoleArn": [
            "arn:aws:iam::account-id:role/emr-on-eks-execution-role"
          ]
        }
      }
    },
    {
      "Sid": "AllowDescribingManagedEndpoint",
      "Effect": "Allow",
      "Action": [
        "emr-containers:DescribeManagedEndpoint"
      ],
      "Resource": [
        "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-cluster-id/endpoints/managed-endpoint-id"
      ]
    }
  ]
}

```

Einschränkungen bei der programmatischen Ausführung

- Pro Konto werden maximal 100 gleichzeitige Ausführungen unterstützt. AWS-Region
- Eine Ausführung wird beendet, wenn sie länger als 30 Tage läuft.
- Die programmatische Ausführung von Notebooks wird mit interaktiven Amazon-EMR-Serverless-Anwendungen nicht unterstützt.

Beispiele für die programmatische Ausführung von EMR-Notebooks

Die folgenden Abschnitte enthalten mehrere Beispiele für die programmatische Ausführung von EMR-Notebooks mit dem AWS CLI Boto3 SDK (Python) und Ruby:

- [Beispiele für Notebook-CLI-Befehle in EMR Studio](#)
- [Python-Beispiele für ein EMR-Notebook](#)
- [Ruby-Beispiele für ein EMR-Notebook](#)

Sie können parametrisierte Notebooks auch als Teil geplanter Workflows mit einem Orchestrierungstool wie Apache Airflow oder Amazon Managed Workflows für Apache Airflow (MWAA) ausführen. Weitere Informationen finden Sie unter [Orchestrieren von Analyseaufträgen auf EMR Notebooks mithilfe von MWAA](#) im AWS -Big-Data-Blog.

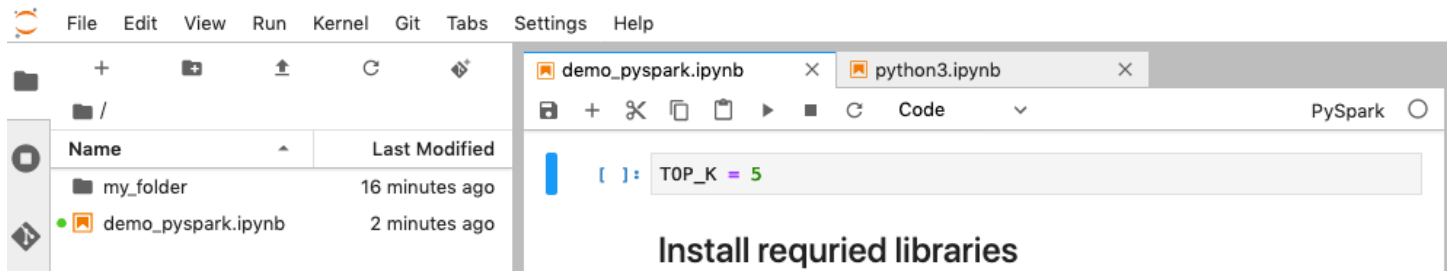
Beispiele für Notebook-CLI-Befehle in EMR Studio

Dieses Thema zeigt CLI-Befehlsbeispiele für ein EMR-Notebook. In dem Beispiel wird das Demo-Notizbuch aus der EMR Notebooks Notebooks-Konsole verwendet. Um das Notebook zu finden, verwenden Sie den Dateipfad relativ zum Home-Verzeichnis. In diesem Beispiel gibt es zwei Notebookdateien, die Sie ausführen können: `demo_pyspark.ipynb` und `my_folder/python3.ipynb`.

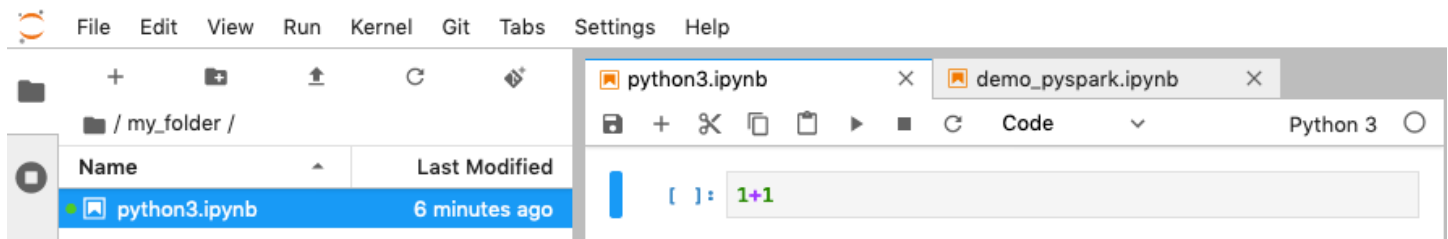
Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Der relative Pfad für die Datei `demo_pyspark.ipynb` ist `demo_pyspark.ipynb`, wie unten dargestellt.



Der relative Pfad für `python3.ipynb` ist `my_folder/python3.ipynb`, wie unten dargestellt.



Informationen zu den API-NotebookExecution-Aktionen in Amazon-EMR finden Sie unter [Amazon-EMR-API-Aktionen](#).

Ein Notebook ausführen

Sie können den verwenden AWS CLI , um Ihr Notebook mit der `start-notebook-execution` Aktion auszuführen, wie die folgenden Beispiele zeigen.

Example — Ausführung eines EMR-Notebooks in einem EMR Studio Workspace mit einem Amazon EMR-Cluster (läuft auf Amazon) EC2

```
aws emr --region us-east-1 \
start-notebook-execution \
--editor-id e-ABCDEFGH123456 \
--notebook-params '{"input_param":"my-value", "good_superhero":["superman", "batman"]}' \
--relative-path test.ipynb \
--notebook-execution-name my-execution \
--execution-engine '{"Id" : "j-1234ABCD123"}' \
--service-role EMR_Notebooks_DefaultRole

{
  "NotebookExecutionId": "ex-ABCDEFGH1234ABCD"
}
```

Example – Ausführen eines EMR-Notebooks in einem EMR Studio Workspace mit einem EMR-Notebooks-Cluster

```
aws emr start-notebook-execution \
  --region us-east-1 \
  --service-role EMR_Notebooks_DefaultRole \
  --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
  --output-notebook-format HTML \
  --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEFGF/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
  --editor-id e-ABCDEFGF \
  --relative-path EMRonEKS-spark_python.ipynb
```

Example – Ausführen eines EMR Notebooks unter Angabe seines Amazon-S3-Standorts

```
aws emr start-notebook-execution \
  --region us-east-1 \
  --notebook-execution-name my-execution-on-emr-on-eks-cluster \
  --service-role EMR_Notebooks_DefaultRole \
  --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
  --output-notebook-format HTML \
  --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEF/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
  --notebook-s3-location '{"Bucket": "amzn-s3-demo-bucket", "Key": "s3-prefix-to-
notebook-location/EMRonEKS-spark_python.ipynb"}' \
  --output-notebook-s3-location '{"Bucket": "amzn-s3-demo-bucket", "Key": "s3-prefix-
for-storing-output-notebook"}'
```

Notebook-Ausgabe

Hier ist die Ausgabe eines Beispiel-Notebooks. Zelle 3 zeigt die neu eingegebenen Parameterwerte.

```

In [1]:
print("Hello world")

Hello world

In [2]: parameters ✕
input_param = "default"
good_superhero = ["batman", "superman"]

In [3]: injected-parameters ✕
# Parameters
good_superhero = ["superman", "batman"]
input_param = "my-value"
new_param = {"nest-key1": "nest-val1", "nest-key2": "nest-val2"}

In [4]:
print(input_param)

my-value

In [5]:
for hero in good_superhero:
    print(hero)

superman
batman

```

Ein Notebook beschreiben

Sie können die `describe-notebook-execution`-Aktion verwenden, um auf Informationen über eine bestimmte Notebook-Ausführung zuzugreifen.

```

aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE

{
  "NotebookExecution": {
    "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
    "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
    "ExecutionEngine": {
      "Id": "j-2QM0V6JAX1TS2",
      "Type": "EMR",
      "MasterInstanceSecurityGroupId": "sg-05ce12e58cd4f715e"
    },
    "NotebookExecutionName": "my-execution",
    "NotebookParams": "{\"input_param\": \"my-value\", \"good_superhero\": [\"superman\", \"batman\"]}",
    "Status": "FINISHED",
    "StartTime": 1593490857.009,
    "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",

```

```

        "LastStateChangeReason": "Execution is finished for cluster j-2QM0V6JAX1TS2.",
        "NotebookInstanceSecurityGroupId": "sg-0683b0a39966d4a6a",
        "Tags": []
    }
}

```

Ein Notebook stoppen

Wenn auf Ihrem Notebook eine Ausführung läuft, die Sie beenden möchten, können Sie dies mit dem `stop-notebook-execution`-Befehl tun.

```

# stop a running execution
aws emr --region us-east-1 \
stop-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T

# describe it
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T

{
  "NotebookExecution": {
    "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
    "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
    "ExecutionEngine": {
      "Id": "j-2QM0V6JAX1TS2",
      "Type": "EMR"
    },
    "NotebookExecutionName": "my-execution",
    "NotebookParams": "{\"input_param\": \"my-value\", \"good_superhero\": \"superman\", \"batman\": \"\"}",
    "Status": "STOPPED",
    "StartTime": 1593490876.241,
    "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:editor-execution/ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
    "LastStateChangeReason": "Execution is stopped for cluster j-2QM0V6JAX1TS2. Internal error",
    "Tags": []
  }
}

```

Listet die Ausführungen für ein Notebook nach Startzeit auf

Sie können einen `--from`-Parameter an `list-notebook-executions` übergeben, um die Ausführungen Ihres Notebooks nach Startzeit aufzulisten.

```
# filter by start time
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000

{
  "NotebookExecutions": [
    {
      "NotebookExecutionId": "ex-IZWZX78UVPAAATC8LHJR129B1RBN4T",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",
      "Status": "STOPPED",
      "StartTime": 1593490876.241
    },
    {
      "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",
      "Status": "RUNNING",
      "StartTime": 1593490857.009
    },
    {
      "NotebookExecutionId": "ex-IZWZYRS0M14L5V95WZ90Q399SKMNW",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",
      "Status": "STOPPED",
      "StartTime": 1593490292.995
    },
    {
      "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",
      "Status": "FINISHED",
      "StartTime": 1593489834.765
    },
    {
      "NotebookExecutionId": "ex-IZWX0ZF88JWDF9J09GJ91R57VI0N",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",

```

```

        "Status": "FAILED",
        "StartTime": 1593488934.688
    }
]
}

```

Listet die Ausführungen für ein Notebook nach Startzeit und Status auf

Der `list-notebook-executions`-Befehl kann auch einen `--status`-Parameter verwenden, um die Ergebnisse zu filtern.

```

# filter by start time and status
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000 --status FINISHED
{
  "NotebookExecutions": [
    {
      "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",
      "Status": "FINISHED",
      "StartTime": 1593490857.009
    },
    {
      "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
      "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
      "NotebookExecutionName": "my-execution",
      "Status": "FINISHED",
      "StartTime": 1593489834.765
    }
  ]
}

```

Python-Beispiele für ein EMR-Notebook

Dieses Thema enthält eine Beispiel-Befehlsdatei. Das Codebeispiel ist eine SDK-Datei für Python (Boto3) namens `demo.py`. Es zeigt die Ausführung des Notebooks. APIs

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen.

Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Informationen zu den Amazon-EMR-API-NotebookExecution-Aktionen finden Sie unter [Amazon-EMR-API-Aktionen](#).

```
import boto3,time

emr = boto3.client(
    'emr',
    region_name='us-west-1'
)

start_resp = emr.start_notebook_execution(
    EditorId='e-40AC8Z06EGGCPJ4DL048KGGGI',
    RelativePath='boto3_demo.ipynb',
    ExecutionEngine={'Id':'j-1HYZS6JQKV11Q'},
    ServiceRole='EMR_Notebooks_DefaultRole'
)

execution_id = start_resp["NotebookExecutionId"]
print(execution_id)
print("\n")

describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)

print(describe_response)
print("\n")

list_response = emr.list_notebook_executions()
print("Existing notebook executions:\n")
for execution in list_response['NotebookExecutions']:
    print(execution)
    print("\n")

print("Sleeping for 5 sec...")
time.sleep(5)

print("Stop execution " + execution_id)
emr.stop_notebook_execution(NotebookExecutionId=execution_id)
```

```
describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)
print(describe_response)
print("\n")
```

Hier ist die Ausgabe vom ausgeführten `demo.py`.

```
ex-IZX56YJDW1D29Q1PHR32WABU2SAPK
```

```
{'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
  'EditorId': 'e-40AC8Z06EGGCPJ4DL048KGGGI', 'ExecutionEngine': {'Id':
  'j-1HYZS6JQKV11Q', 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STARTING',
  'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()),
  'Arn': 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
  IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is starting
  for cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId':
  '70f12c5f-1dda-45b7-adf6-964987d373b7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-
  amzn-requestid': '70f12c5f-1dda-45b7-adf6-964987d373b7', 'content-type': 'application/
  x-amz-json-1.1', 'content-length': '448', 'date': 'Wed, 19 Aug 2020 00:49:22 GMT'},
  'RetryAttempts': 0}}
```

Existing notebook executions:

```
{'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK', 'EditorId':
  'e-40AC8Z06EGGCPJ4DL048KGGGI', 'NotebookExecutionName': '', 'Status': 'STARTING',
  'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal())}
```

```
{'NotebookExecutionId': 'ex-IZX5ABS5PR1E5AHMFYEMX3JJIORRB', 'EditorId':
  'e-40AC8Z06EGGCPJ4DL048KGGGI', 'NotebookExecutionName': '', 'Status': 'RUNNING',
  'StartTime': datetime.datetime(2020, 8, 19, 0, 48, 36, 373000, tzinfo=tzlocal())}
```

```
{'NotebookExecutionId': 'ex-IZX5GLVXIU1HNI8BWVW057F6MF4VE', 'EditorId':
  'e-40AC8Z06EGGCPJ4DL048KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
  'StartTime': datetime.datetime(2020, 8, 19, 0, 45, 14, 646000, tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 8, 19, 0, 46, 26, 543000, tzinfo=tzlocal())}
```

```
{'NotebookExecutionId': 'ex-IZX5CV8YDU08JAIWMXN2VH32RUIT1', 'EditorId':
  'e-40AC8Z06EGGCPJ4DL048KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
  'StartTime': datetime.datetime(2020, 8, 19, 0, 43, 5, 807000, tzinfo=tzlocal()),
  'EndTime': datetime.datetime(2020, 8, 19, 0, 44, 31, 632000, tzinfo=tzlocal())}
```

```
{'NotebookExecutionId': 'ex-IZX5AS0PPW55CEEURZ9NS0WSUJZ6', 'EditorId':
'e-40AC8Z06EGGCPJ4DL048KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
'StartTime': datetime.datetime(2020, 8, 19, 0, 42, 29, 265000, tzinfo=tzlocal()),
'EndTime': datetime.datetime(2020, 8, 19, 0, 43, 48, 320000, tzinfo=tzlocal())}
```

```
{'NotebookExecutionId': 'ex-IZX57YF5Q53BKWLR4I5QZ14HJ7DRS', 'EditorId':
'e-40AC8Z06EGGCPJ4DL048KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
'StartTime': datetime.datetime(2020, 8, 19, 0, 38, 37, 81000, tzinfo=tzlocal()),
'EndTime': datetime.datetime(2020, 8, 19, 0, 40, 39, 646000, tzinfo=tzlocal())}
```

Sleeping for 5 sec...

Stop execution ex-IZX56YJDW1D29Q1PHR32WABU2SAPK

```
{'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
'EditorId': 'e-40AC8Z06EGGCPJ4DL048KGGGI', 'ExecutionEngine': {'Id':
'j-1HYZS6JQKV11Q', 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STOPPING',
'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()),
'Arn': 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is being stopped
for cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId':
'2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-
amzn-requestid': '2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'content-type': 'application/
x-amz-json-1.1', 'content-length': '453', 'date': 'Wed, 19 Aug 2020 00:49:30 GMT'},
'RetryAttempts': 0}}}
```

Ruby-Beispiele für ein EMR-Notebook

Dieses Thema enthält ein Ruby-Beispiel, das die Funktionalität eines Notebooks demonstriert.

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Die folgenden Ruby-Codebeispiele demonstrieren die Verwendung der Notebook-Ausführungs-API.

```
# prepare an Amazon EMR client

emr = Aws::EMR::Client.new(
  region: 'us-east-1',
  access_key_id: 'AKIA...JKPKA',
  secret_access_key: 'rLMeu...vU00LrAC1',
)
```

Starten der Notebook-Ausführung und Abrufen der Ausführungs-ID

In diesem Beispiel sind der Amazon-S3-Editor und das EMR-Notebook `s3://amzn-s3-demo-bucket/notebooks/e-EA8VGAA429FEQTC8HC9ZHWISK/test.ipynb`.

Informationen zu den Amazon-EMR-API-NotebookExecution-Aktionen finden Sie unter [Amazon-EMR-API-Aktionen](#).

```
start_response = emr.start_notebook_execution({
  editor_id: "e-EA8VGAA429FEQTC8HC9ZHWISK",
  relative_path: "test.ipynb",

  execution_engine: {id: "j-3U82I95AMALGE"},

  service_role: "EMR_Notebooks_DefaultRole",
})

notebook_execution_id = start_resp.notebook_execution_id
```

Beschreibung der Ausführung des Notebooks und Ausdrucken der Details

```
describe_resp = emr.describe_notebook_execution({
  notebook_execution_id: notebook_execution_id
})
puts describe_resp.notebook_execution
```

Die Ausgabe der obigen Befehle wird wie folgt aussehen.

```
{
 :notebook_execution_id=>"ex-IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
 :editor_id=>"e-EA8VGAA429FEQTC8HC9ZHWISK",
```

```

:execution_engine=>{:id=>"j-3U82I95AMALGE", :type=>"EMR", :master_instance_security_group_id=>n
:notebook_execution_name=>"",
:notebook_params=>nil,
:status=>"STARTING",
:start_time=>2020-07-23 15:07:07 -0700,
:end_time=>nil,
:arn=>"arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZX3VTVZWVWP27KUB90BZ7V9IEDG",
:output_notebook_uri=>nil,
:last_state_change_reason=>"Execution is starting for cluster
j-3U82I95AMALGE.", :notebook_instance_security_group_id=>nil,
:tags=>[]
}

```

Notebookfilter

```

"EditorId": "e-XXXX",           [Optional]
"From" : "1593400000.000",      [Optional]
"To" :

```

Beenden der Notebook-Ausführung

```

stop_resp = emr.stop_notebook_execution({
  notebook_execution_id: notebook_execution_id
})

```

Aktivieren des Identitätswechsels zur Überwachung von Spark-Benutzer- und -Aufgabenaktivitäten

EMR-Notebooks ermöglichen die Konfiguration von Benutzer-Identitätswechseln auf einem Spark-Cluster. Mit dieser Funktion können Sie die Auftragsaktivität nachverfolgen, die innerhalb des Notebook-Editors initiiert wurde. Darüber hinaus verfügen EMR-Notebooks über ein integriertes Jupyter-Notebook-Widget zur Anzeige von Details zu Spark-Aufgaben zusammen mit der Abfrageausgabe im Notebook-Editor. Das Widget ist standardmäßig verfügbar und erfordert keine spezielle Konfiguration. Um die Verlaufsserver anzeigen zu können, muss Ihr Client jedoch so konfiguriert sein, dass Amazon-EMR-Webschnittstellen angezeigt werden, die auf dem Primärknoten gehostet werden.

 Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Einrichten der Spark-Benutzerkennung

Standardmäßig stammen Spark-Aufträge, die Benutzer mit dem Notebook-Editor übermitteln, scheinbar aus einer unbestimmten `livy`-Benutzeridentität. Sie können eine Benutzerkennung für den Cluster konfigurieren, damit diese Aufträge stattdessen mit der Benutzeridentität verknüpft werden, die den Code ausgeführt hat. HDFS-Benutzerverzeichnisse auf dem Primärknoten werden für jede Benutzeridentität erstellt, die Code im Notebook ausführt. Beispiel: Wenn der Benutzer `NbUser1` Code aus dem Notebook-Editor ausführt, können Sie eine Verbindung mit dem Primärknoten herstellen und sehen, dass `hadoop fs -ls /user` das Verzeichnis `/user/user_NbUser1` zeigt.

Sie können diese Funktion aktivieren, indem Sie Eigenschaften in den Konfigurationsklassifizierungen `core-site` und `livy-conf` festlegen. Dieses Feature ist nicht standardmäßig verfügbar, wenn Amazon EMR einen Cluster zusammen mit einem Notebook erstellt. Weitere Informationen zur Verwendung von Konfigurationsklassifizierungen zum Anpassen von Anwendungen finden Sie unter [Konfigurieren von Anwendungen](#) in Amazon-EMR-Versionshinweise.

Verwenden Sie die folgenden Konfigurationsklassifizierungen und Werte, um einen Benutzer-Identitätswechsel für EMR Notebooks: zu aktivieren:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "hadoop.proxyuser.livy.groups": "*",
      "hadoop.proxyuser.livy.hosts": "*"
    }
  },
  {
    "Classification": "livy-conf",
```

```
    "Properties": {  
      "livy.impersonation.enabled": "true"  
    }  
  }  
]
```

Verwenden des Spark-Widgets für die Auftragsüberwachung

Wenn Sie im Notebook-Editor Code ausführen, der Spark-Aufträge im EMR-Cluster ausführt, enthält die Ausgabe ein Jupyter-Notebook-Widget für die Spark-Auftragsüberwachung. Das Widget stellt Auftragsdetails, nützliche Links zur Spark-Verlaufsserverseite und zur Hadoop-Auftragsverlaufsseite sowie praktische Links zu Auftragsprotokollen in Amazon S3 für alle fehlgeschlagenen Aufträge bereit.

Um Verlaufsserverseiten auf dem Cluster-Primärknoten anzuzeigen, müssen Sie einen SSH-Client und einen Proxy nach Bedarf einrichten. Weitere Informationen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#). Um Protokolle in Amazon S3 anzuzeigen, muss die Cluster-Protokollierung aktiviert sein. Dies ist die Standardeinstellung für neue Cluster. Weitere Informationen finden Sie unter [In Amazon S3 archivierte Protokolldateien anzeigen](#).

Nachstehend finden Sie ein Beispiel für die Spark-Auftragsüberwachung.

Spark Job Progress

Click to expand and view Spark job details

Job [0]: reduce at <stdin>:16

Progress for reduce at <stdin>:16 Job Progress: 16/16 Tasks Comp...

Stage [ID]: name at [source]:[line]	Status	Task Progress	Elapsed Time (seconds)	Failed Task Logs
Stage [0]: coalesce at Natl...java:0	COMPLETE	4/4	11.71	
Stage [1]: reduce at <stdin>:16	COMPLETE	12/12		

Job [1]: foreach at <stdin>:24

Progress for foreach at <stdin>:24 Job Progress: 4/12 Tasks Complete

Stage [ID]: name at [source]:[line]	Status	Task Progress	Elapsed Time (seconds)	Failed Task Logs
Stage [2]: coalesce at Natl...java:0	SKIPPED	0/4	n/a	
Stage [3]: foreach at <stdin>:24	FAILED	4/12	1.212	stderr stdout

For failed jobs, click these links to view logs in Amazon S3 when logging is enabled on the cluster.

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1542497924776_0001	pyspark	idle	Link	Link	✓

SparkSession available as 'spark'.

An error occurred while calling z... apache.spark.api.python.Python... collectAndServe.
 : org.apache.spark.SparkException: Job aborted due to stage failure: Task 3.0 failed 4 times, most recent failure
 e: Loss of connection to the ResourceManager at: ip-172-31-20-106.ec2.internal, executionId=org.apache.spark.api.python.PythonExcepti
 on: Tr...
 File /mnt/yarn/usercache/user_jeffgoll/appcache/application_1542497924776_0002/pyspark.zip/pyspark/worker.py, line 248, in process
 main
 File /mnt/yarn/usercache/user_jeffgoll/appcache/application_1542497924776_0002/pyspark.zip/pyspark/worker.py, line 248, in process
 serializer.dump_stream(func(split_index, iterator), outfile)
 File /usr/lib/spark/python/lib/pyspark.zip/pyspark/rdd.py, line 2440, in pipeline_func
 File /usr/lib/spark/python/lib/pyspark.zip/pyspark/rdd.py, line 2440, in pipeline_func

Click this link to view Spark History Server.

Click this link to view Hadoop Job History.

Sicherheit und Zugriffskontrolle für EMR Notebooks

Es sind verschiedene Features verfügbar, mit denen Sie die Sicherheitslage von EMR Notebooks anpassen können. Dadurch wird sichergestellt, dass nur autorisierte Benutzer Zugriff auf ein EMR Notebook haben, mit Notebooks arbeiten und den Notebook-Editor zum Ausführen von Code auf dem Cluster verwenden können. Diese Features arbeiten mit den für Amazon EMR und Amazon-EMR-Cluster verfügbaren Sicherheitsfunktionen zusammen. Weitere Informationen finden Sie unter [Sicherheit in Amazon EMR](#).

- Sie können AWS Identity and Access Management Richtlinienenerklärungen zusammen mit Notebook-Tags verwenden, um den Zugriff einzuschränken. Weitere Informationen erhalten Sie unter [Funktionsweise von Amazon EMR mit IAM](#) und [Beispielhafte identitätsbasierte Richtlinienanweisungen für EMR Notebooks](#).

- EC2 Amazon-Sicherheitsgruppen agieren als virtuelle Firewalls, die den Netzwerkverkehr zwischen der primären Instance des Clusters und dem Notebook-Editor steuern. Sie können Standardwerte verwenden oder diese Sicherheitsgruppen anpassen. Weitere Informationen finden Sie unter [EC2 Sicherheitsgruppen für EMR Notebooks angeben](#).
- Sie geben eine AWS Servicerolle an, die bestimmt, welche Berechtigungen ein EMR-Notebook bei der Interaktion mit anderen AWS Diensten hat. Weitere Informationen finden Sie unter [Servicerolle für EMR Notebooks](#).

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Installation und Verwendung von Kernen und Bibliotheken in EMR Studio

Jedes EMR Notebook wird mit einer Reihe vorinstallierter Bibliotheken und Kernel ausgeliefert. Sie können in einem EMR-Cluster zusätzliche Bibliotheken und Kernel installieren, wenn der Cluster Zugriff auf das Repository hat, in dem sich die Kernel und Bibliotheken befinden. Beispielsweise müssen Sie für Cluster in privaten Subnetzen möglicherweise die Netzwerkadressübersetzung (Network Address Translation, NAT) konfigurieren und dem Cluster einen Pfad für den Zugriff auf das öffentliche PyPI-Repository angeben, um eine Bibliothek zu installieren. Weitere Informationen zum Konfigurieren des externen Zugriffs für verschiedene Netzwerkkonfigurationen finden Sie unter [Szenarien und Beispiele](#) im Amazon-VPC-Benutzerhandbuch.

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer

zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Serverlose EMR-Anwendungen verfügen über die folgenden vorinstallierten Bibliotheken für Python und: PySpark

- Python-Bibliotheken — ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, scipy
- PySpark Bibliotheken — ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, scipy

Installieren von Kernels und Python-Bibliotheken auf einem Cluster-Primärknoten

Mit Amazon-EMR-Version 5.30.0 und höher (außer Version 6.0.0) können Sie zusätzliche Python-Bibliotheken und Kernel auf dem Primärknoten des Clusters installieren. Nach der Installation stehen diese Kernel und Bibliotheken allen Benutzern zur Verfügung, die ein an den Cluster angefügtes EMR Notebook ausführen. Auf diese Weise installierte Python-Bibliotheken sind nur für Prozesse verfügbar, die auf dem Primärknoten ausgeführt werden. Die Bibliotheken werden nicht auf Core- oder Aufgabenknoten installiert und sind für Executors, die auf diesen Knoten ausgeführt werden, nicht verfügbar.

Note

Für die Amazon-EMR-Versionen 5.30.1, 5.31.0 und 6.1.0 müssen Sie zusätzliche Schritte unternehmen, um Kernel und Bibliotheken auf dem Primärknoten eines Clusters zu installieren.

Um das Feature zu aktivieren, gehen Sie wie folgt vor:

1. Stellen Sie sicher, dass die der Servicerolle für EMR Notebooks zugeordnete Berechtigungsrichtlinie die folgende Aktion zulässt:

```
elasticmapreduce:ListSteps
```

Weitere Informationen finden Sie unter [Servicerolle für EMR-Notebooks](#).

2. Verwenden Sie den AWS CLI , um einen Schritt auf dem Cluster auszuführen, der EMR Notebooks einrichtet, wie im folgenden Beispiel gezeigt. Sie müssen den Schrittnamen EMRNotebooksSetup verwenden. *us-east-1* Ersetzen Sie durch die Region, in der sich

Ihr Cluster befindet. Weitere Informationen finden Sie unter [Hinzufügen von Schritten zu einem Cluster AWS CLI](#).

```
aws emr add-steps --cluster-id MyClusterID --steps
  Type=CUSTOM_JAR,Name=EMRNotebooksSetup,ActionOnFailure=CONTINUE,Jar=s3://us-east-1.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://
  awssupportdatasvcs.com/bootstrap-actions/EMRNotebooksSetup/emr-notebooks-
  setup.sh"]
```

Sie können Kernel und Bibliotheken mithilfe von pip oder conda im `/emr/notebook-env/bin-`Verzeichnis auf dem Primärknoten installieren.

Example – Installieren von Python-Bibliotheken

Führen Sie im Python3-Kernel den `%pip` Magic als Befehl in einer Notebook-Zelle aus, um Python-Bibliotheken zu installieren.

```
%pip install pmdarima
```

Möglicherweise müssen Sie den Kernel neu starten, um aktualisierte Pakete verwenden zu können. Sie können auch die `%%sh`-Spark-Magic zum Aufrufen von pip verwenden.

```
%%sh
/emr/notebook-env/bin/pip install -U matplotlib
/emr/notebook-env/bin/pip install -U pmdarima
```

Wenn Sie einen PySpark Kernel verwenden, können Sie entweder Bibliotheken auf dem Cluster mithilfe von pip Befehlen installieren oder Bibliotheken für Notebooks innerhalb eines Notebooks verwenden. PySpark

Um pip-Befehle auf dem Cluster vom Terminal aus auszuführen, stellen Sie zunächst über SSH eine Verbindung zum Primärknoten her, wie die folgenden Befehle zeigen.

```
sudo pip3 install -U matplotlib
sudo pip3 install -U pmdarima
```

Alternativ können Sie Bibliotheken im Notebookbereich verwenden. Bei Bibliotheken für Notebooks ist Ihre Bibliotheksinstallation auf den Umfang Ihrer Sitzung beschränkt und erfolgt auf allen Spark-Executoren. Weitere Informationen finden Sie unter [Verwenden von Notebook Bibliotheken](#).

Wenn Sie mehrere Python-Bibliotheken in einen PySpark Kernel packen möchten, können Sie auch eine isolierte virtuelle Python-Umgebung erstellen. Beispiele finden Sie unter [Verwenden Virtualenv](#).

Um eine virtuelle Python-Umgebung in einer Sitzung zu erstellen, verwenden Sie die Spark-Eigenschaft `spark.yarn.dist.archives` aus dem `%%configure` magischen Befehl in der ersten Zelle in einem Notebook, wie das folgende Beispiel zeigt.

```
%%configure -f
{
  "conf": {
    "spark.yarn.appMasterEnv.PYSPARK_PYTHON": "./environment/bin/python",
    "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON": "./environment/bin/python",
    "spark.yarn.dist.archives": "s3://amzn-s3-demo-bucket/prefix/
my_pyspark_venv.tar.gz#environment",
    "spark.submit.deployMode": "cluster"
  }
}
```

Auf ähnliche Weise können Sie eine Spark-Executor-Umgebung erstellen.

```
%%configure -f
{
  "conf": {
    "spark.yarn.appMasterEnv.PYSPARK_PYTHON": "./environment/bin/python",
    "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON": "./environment/bin/python",
    "spark.executorEnv.PYSPARK_PYTHON": "./environment/bin/python",
    "spark.yarn.dist.archives": "s3://amzn-s3-demo-bucket/prefix/
my_pyspark_venv.tar.gz#environment",
    "spark.submit.deployMode": "cluster"
  }
}
```

Sie können es auch `conda` zur Installation von Python-Bibliotheken verwenden. Für die Verwendung von `conda` benötigen Sie keinen Sudo-Zugriff. Sie müssen über SSH eine Verbindung mit dem Primärknoten herstellen und dann `conda` vom Terminal aus ausführen. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Example – Installieren eines Kernels

Das folgende Beispiel zeigt die Installation des Kotlin-Kernels mithilfe eines Terminalbefehls, während eine Verbindung zum Primärknoten eines Clusters besteht:

```
sudo /emr/notebook-env/bin/conda install kotlin-jupyter-kernel -c jetbrains
```

Note

Diese Anweisungen installieren keine Kernel-Abhängigkeiten. Wenn Ihr Kernel Abhängigkeiten von Drittanbietern hat, müssen Sie möglicherweise zusätzliche Einrichtungsschritte durchführen, bevor Sie den Kernel mit Ihrem Notebook verwenden können.

Überlegungen und Einschränkungen bei Bibliotheken für Notebooks

Beachten Sie bei der Verwendung von Bibliotheken im Format Notebook-Scoped Folgendes:

- Bibliotheken für Notebooks sind für Cluster verfügbar, die Sie mit Amazon-EMR-Versionen 5.26.0 und höher erstellen.
- Bibliotheken für Notebooks sind nur für die Verwendung mit dem Kernel vorgesehen. PySpark
- Jeder Benutzer kann zusätzliche dedizierte Notebook-Bibliotheken innerhalb einer Notebook-Zelle installieren. Diese Bibliotheken stehen diesem Notebook-Benutzer nur während genau einer Notebook-Sitzung zur Verfügung. Wenn andere Benutzer dieselben Bibliotheken benötigen oder derselbe Benutzer dieselben Bibliotheken in einer anderen Sitzung benötigt, muss die Bibliothek neu installiert werden.
- Sie können nur die Bibliotheken deinstallieren, die mit der `install_pypi_package`-API installiert wurden. Sie können keine Bibliotheken deinstallieren, die auf dem Cluster vorinstalliert sind.
- Wenn dieselben Bibliotheken mit unterschiedlichen Versionen auf dem Cluster und als Notebook-Bibliotheken installiert sind, überschreibt die Version der Notebook-Bibliothek die Version der Cluster-Bibliothek.

Arbeiten mit Notebook-Bibliotheken

Um Bibliotheken zu installieren, muss Ihr Amazon-EMR-Cluster Zugriff auf das PyPI-Repository haben, in dem sich die Bibliotheken befinden.

Die folgenden Beispiele zeigen einfache Befehle zum Auflisten, Installieren und Deinstallieren von Bibliotheken innerhalb einer Notebookzelle mithilfe des Kernels und. PySpark APIs Weitere Beispiele

finden [Sie im Beitrag Installieren von Python-Bibliotheken auf einem laufenden Cluster mit EMR Notebooks](#) im AWS Big Data-Blog.

Example – Auflisten aktueller Bibliotheken

Der folgende Befehl listet die Python-Pakete auf, die für die aktuelle Spark-Notebook-Sitzung verfügbar sind. Hiermit werden Bibliotheken aufgelistet, die auf dem Cluster installiert sind, und Bibliotheken für Notebook-Bereiche.

```
sc.list_packages()
```

Example – Installieren der Celery-Bibliothek

Mit dem folgenden Befehl wird die [Celery](#)-Bibliothek als Notebook-Bibliothek installiert.

```
sc.install_pypi_package("celery")
```

Nach der Installation der Bibliothek bestätigt der folgende Befehl, dass die Bibliothek auf dem Spark-Treiber und den Executors verfügbar ist.

```
import celery
sc.range(1,10000,1,100).map(lambda x: celery.__version__).collect()
```

Example – Installieren der Arrow-Bibliothek unter Angabe der Version und des Repositorys

Mit dem folgenden Befehl wird die [Pfeilbibliothek](#) als Bibliothek mit einem „Notebook-Scoped“-Format mit einer Spezifikation der Bibliotheksversion und der Repository-URL installiert.

```
sc.install_pypi_package("arrow==0.14.0", "https://pypi.org/simple")
```

Example – Deinstallieren einer Bibliothek

Der folgende Befehl deinstalliert die Pfeilbibliothek und entfernt sie als Notebook-Bibliothek aus der aktuellen Sitzung.

```
sc.uninstall_package("arrow")
```

Verknüpfen von Git-basierten Repositorys mit EMR Notebooks

Sie können Git-basierte Repositorys mit Ihren Amazon-EMR-Notebooks verknüpfen, um Ihre Notebooks in einer versionskontrollierten Umgebung zu speichern. Sie können einem Notebook bis zu drei Repositorys zuordnen. Folgende Git-basierte Services werden unterstützt:

- [AWS CodeCommit](#)
- [GitHub](#)
- [Bitbucket](#)
- [GitLab](#)

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Die Verknüpfung von Git-basierten Repositorys mit Ihrem Notebook hat folgende Vorteile:

- Versionskontrolle – Sie können Codeänderungen in einem Versionskontrollsystem aufzeichnen, damit Sie den Verlauf Ihrer Änderungen überprüfen und selektiv rückgängig machen können.
- Zusammenarbeit – Kollegen, die in verschiedenen Notebooks arbeiten, können Code über Git-basierte Remote-Repositorys füreinander freigeben. Notebooks können Code aus Remote-Repositorys klonen oder zusammenführen und Änderungen in diese Remote-Repositorys zurückübertragen.
- Wiederverwendung von Code — Viele Jupyter-Notebooks, die Datenanalyse oder Techniken des maschinellen Lernens demonstrieren, sind in öffentlich gehosteten Repositorys verfügbar, wie z. GitHub. Sie können Ihre Notebooks mit einem Repository verknüpfen, um die in diesem Repository enthaltenen Jupyter-Notebooks wiederzuverwenden.

Um Git-basierte Repositorys mit EMR Notebooks verwenden zu können, fügen Sie die Repositorys als Ressourcen in der Amazon-EMR-Konsole hinzu, ordnen Sie Anmeldeinformationen für

Repositorys zu, die eine Authentifizierung erfordern, und verknüpfen Sie sie mit Ihren Notebooks. In der Amazon-EMR-Konsole können Sie eine Liste der Repositorys, die in Ihrem Konto gespeichert sind, sowie Details zu den einzelnen Repositorys anzeigen. Sie können ein vorhandenes Git-basiertes Repository mit einem Notebook verknüpfen, wenn Sie es erstellen.

Themen

- [Voraussetzungen und Überlegungen bei der Integration eines EMR-Notebooks in ein Repository](#)
- [Hinzufügen eines Git-basierten Repositorys zu Amazon EMR](#)
- [Ein Git-basiertes Repository aus einem EMR Studio Workspace aktualisieren oder löschen](#)
- [Ein Git-basiertes Repository in EMR Studio verknüpfen oder die Verknüpfung aufheben](#)
- [Erstellen Sie ein neues Notebook mit einem zugehörigen Git-Repository in EMR Studio](#)
- [Verwenden Sie Git-Repositorys in einem EMR Studio-Notebook](#)

Voraussetzungen und Überlegungen bei der Integration eines EMR-Notebooks in ein Repository

Beachten Sie die folgenden bewährten Methoden in Bezug auf Commits, Berechtigungen und Hosting, wenn Sie planen, ein Git-basiertes Repository in EMR Notebooks zu integrieren.

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

AWS CodeCommit

Wenn Sie ein CodeCommit Repository verwenden, müssen Sie Git-Anmeldeinformationen und HTTPS mit verwenden CodeCommit. SSH-Schlüssel und HTTPS mit dem AWS CLI Credential Helper werden nicht unterstützt. CodeCommit unterstützt keine persönlichen Zugriffstoken (PATs). Weitere Informationen finden Sie unter [Verwenden von IAM mit CodeCommit: Git-Anmeldeinformationen, SSH-Schlüsseln und AWS Zugriffsschlüsseln](#) im IAM-Benutzerhandbuch

und [Einrichtung für HTTPS-Benutzer mit Git-Anmeldeinformationen](#) im AWS CodeCommit Benutzerhandbuch.

Überlegungen zu Zugriff und Berechtigungen

Bevor Sie ein Repository mit Ihrem Notebook verknüpfen, müssen Sie sicherstellen, dass Ihr Cluster, Ihre IAM-Rolle für EMR Notebooks und Ihre Sicherheitsgruppen über die richtigen Einstellungen und Berechtigungen verfügen. Sie können auch Git-basierte Repositories konfigurieren, die Sie in einem privaten Netzwerk hosten, indem Sie den Anweisungen unter [Ein privat gehostetes Git-Repository für EMR Notebooks konfigurieren](#) folgen.

- Cluster-Internetzugriff – Die Netzwerkschnittstelle, die gestartet wird, hat nur eine private IP-Adresse. Das bedeutet, dass der Cluster, mit dem Ihr Notebook eine Verbindung herstellt, sich in einem privaten Subnetz mit einem NAT-Gateway (Network Address Translation) befinden oder über ein Virtual Private Gateway auf das Internet zugreifen können muss. Weitere Informationen finden Sie unter [Amazon-VPC-Optionen](#).

Die Sicherheitsgruppen für Ihr Notebook müssen eine Regel für ausgehenden Datenverkehr enthalten, sodass das Notebook Datenverkehr vom Cluster an das Internet weiterleiten kann. Wir empfehlen, eigene Sicherheitsgruppen zu erstellen. Weitere Informationen finden Sie unter [EC2 Sicherheitsgruppen für EMR Notebooks angeben](#).

Important

Wenn die Netzwerkschnittstelle in ein öffentliches Subnetz gestartet wird, kann sie nicht über ein Internet-Gateway (IGW) mit dem Internet kommunizieren.

- Berechtigungen für AWS Secrets Manager — Wenn Sie Secrets Manager zum Speichern von Geheimnissen verwenden, die Sie für den Zugriff auf ein Repository verwenden, [the section called “EMR-Notebooks-Rolle”](#) muss eine Berechtigungsrichtlinie angehängt sein, die die `secretsmanager:GetSecretValue` Aktion ermöglicht.

Ein privat gehostetes Git-Repository für EMR Notebooks konfigurieren

Verwenden Sie die folgenden Anweisungen, um privat gehostete Repositories für EMR Notebooks zu konfigurieren. Sie müssen eine Konfigurationsdatei mit Informationen zu Ihren DNS- und Git-Servern bereitstellen. Amazon EMR verwendet diese Informationen, um EMR Notebooks zu konfigurieren, die den Datenverkehr an Ihre privat gehosteten Repositories weiterleiten können.

Voraussetzungen

Bevor Sie ein privat gehostetes Git-Repository für EMR Notebooks konfigurieren, benötigen Sie Folgendes:

- Ein Amazon S3 Control Ort, an dem Dateien für Ihr EMR-Notizbuch gespeichert werden.

Um ein oder mehrere privat gehostete Git-Repositorys für EMR Notebooks zu konfigurieren

1. Erstellen Sie eine Konfigurationsdatei mit der bereitgestellten Vorlage. Geben Sie für jeden Git-Server, den Sie in Ihrer Konfiguration angeben möchten, die folgenden Werte an:

- **DnsServerIPv4**- Die IPv4 Adresse Ihres DNS-Servers. Wenn Sie Werte für sowohl DnsServerIPv4 als auch GitServerIPv4List angeben, hat der Wert für DnsServerIPv4 Vorrang und wird zur Auflösung Ihres GitServerDnsName verwendet.

Note

Um privat gehostete Git-Repositorys verwenden zu können, muss Ihr DNS-Server eingehenden Zugriff von EMR Notebooks zulassen. Wir empfehlen Ihnen dringend, Ihren DNS-Server vor anderen, unbefugten Zugriffen zu schützen.

- **GitServerDnsName** – Der DNS-Name Ihres Git-Servers. Zum Beispiel "git.example.com".
- **GitServerIPv4List**- Eine Liste von IPv4 Adressen, die zu deinen Git-Servern gehören.

```
[
  {
    "Type": "PrivatelyHostedGitConfig",
    "Value": [
      {
        "DnsServerIPv4": "<10.24.34.xxx>",
        "GitServerDnsName": "<enterprise.git.com>",
        "GitServerIPv4List": [
          "<xxx.xxx.xxx.xxx>",
          "<xxx.xxx.xxx.xxx>"
        ]
      },
      {
```

```

        "DnsServerIPv4": "<10.24.34.xxx>",
        "GitServerDnsName": "<git.example.com>",
        "GitServerIPv4List": [
            "<xxx.xxx.xxx.xxx>",
            "<xxx.xxx.xxx.xxx>"
        ]
    }
}
]

```

2. Speichern Sie Ihre Konfigurationsdatei unter `configuration.json`.
3. Laden Sie die Konfigurationsdatei in den von Ihnen angegebenen Amazon-S3-Speicherort in einem Ordner mit dem Namen `life-cycle-configuration` hoch. Wenn Ihr Standard-S3-Speicherort beispielsweise `s3://amzn-s3-demo-bucket/notebooks` lautet, sollte sich Ihre Konfigurationsdatei unter `s3://amzn-s3-demo-bucket/notebooks/life-cycle-configuration/configuration.json` befinden.

Important

Wir empfehlen dringend, den Zugriff auf Ihren `life-cycle-configuration`-Ordner nur auf Ihre EMR-Notebooks-Administratoren und auf die Servicerolle für EMR Notebooks zu beschränken. Sie sollten auch `configuration.json` vor unbefugtem Zugriff schützen. Anweisungen finden Sie unter [Steuern des Zugriffs auf einen Bucket mit Benutzerrichtlinien](#) oder [Bewährte Sicherheitsmethoden für Amazon S3](#).

Anweisungen zum Hochladen finden Sie unter [Erstellen eines Ordners](#) und [Hochladen von Objekten](#) im Benutzerhandbuch für Amazon Simple Storage Service.

Hinzufügen eines Git-basierten Repositorys zu Amazon EMR

In den folgenden Abschnitten finden Sie Informationen zum Hinzufügen eines Git-basierten Repositorys zu einem EMR-Notebook in der alten Konsole oder zu einem EMR Studio Workspace in der Konsole.

 Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Console

Da EMR Notebooks in der neuen Konsole EMR-Studio-Workspaces sind, können Sie den Anweisungen unter [Git-basierte Repositories mit einem EMR Studio Workspace verknüpfen](#) folgen, um Ihrem Workspace bis zu drei Git-Repositories zuzuordnen.

Alternativ können Sie den JupyterLab Git Erweiterung. Wählen Sie das Git-Symbol in der linken Seitenleiste Ihres Jupyterlab-Notebooks, um auf die Erweiterung zuzugreifen. Informationen zur Erweiterung finden Sie im [GitHub jupyterlab-git](#) Repo.

Um ein Git-Repository mit einem Workspace zu verknüpfen, muss Ihr Studio-Administrator Schritte unternehmen, um das Studio so zu konfigurieren, dass die Verknüpfung mit Git-Repositories zulässig ist. Weitere Informationen finden Sie unter [Zugriff und Berechtigungen für Git-basierte Repositories einrichten](#).

Ein Git-basiertes Repository aus einem EMR Studio Workspace aktualisieren oder löschen

In den folgenden Abschnitten finden Sie Informationen zum Löschen eines Git-basierten Repositories aus einem EMR-Notebook in der alten Konsole oder aus einem EMR Studio Workspace in der Konsole.

 Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer

zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Console

Da es sich bei EMR Notebooks in der neuen Konsole um EMR-Studio-Workspaces handelt, finden Sie weitere Informationen zur Arbeit mit Git-Repositorys in Ihrem Workspace unter [Git-basierte Repositorys mit einem EMR Studio Workspace verknüpfen](#). Derzeit können Sie Git-Repositorys jedoch nicht aus Workspaces löschen.

Ein Git-basiertes Repository in EMR Studio verknüpfen oder die Verknüpfung aufheben

Gehen Sie wie folgt vor, um ein Git-basiertes Repository mit einem EMR-Notebook in der alten Konsole oder mit einem EMR Studio-Workspace in der Konsole zu verknüpfen oder die Verknüpfung aufzuheben.

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Console

Da es sich bei EMR Notebooks in der neuen Konsole um EMR-Studio-Workspaces handelt, finden Sie weitere Informationen zur Arbeit mit Git-Repositorys in Ihrem Workspace unter [Git-basierte Repositorys mit einem EMR Studio Workspace verknüpfen](#). Derzeit können Sie Git-Repositorys jedoch nicht aus Workspaces löschen.

Grundlegendes zum Repository-Status

Ein Git-Repository kann beliebige der folgenden Status in der Repository-Liste aufweisen. Für weitere Informationen zum Verknüpfen von EMR-Notebooks mit Git-Repositorys siehe [Ein Git-basiertes Repository in EMR Studio verknüpfen oder die Verknüpfung aufheben](#).

Status	Bedeutung
Verknüpfen	Das Git-Repository wird mit dem Notebook verknüpft. Während der Status des Repositorys Linking (Verknüpfung wird hergestellt) lautet, können Sie das Notebook anhalten.
Verknüpft	Das Git-Repository ist mit dem Notebook verknüpft. Solange das Repository den Status Linked (Verknüpft) aufweist, ist es mit dem Remote-Repository verbunden.
Verknüpfung fehlgeschlagen	Das Git-Repository konnte nicht mit dem Notebook verknüpft werden. Sie können erneut versuchen, es zu verknüpfen.
Verknüpfung aufheben	Die Verknüpfung des Git-Repositorys mit dem Notebook wird aufgehoben. Während der Status des Repositorys Unlinking (Verknüpfung wird aufgehoben) lautet, können Sie das Notebook nicht anhalten. Durch das Aufheben der Verknüpfung eines Git-Repositorys mit einem Notebook wird nur die Verbindung mit dem Remote-Repository getrennt; es wird kein Code aus dem Notebook gelöscht.
Verknüpfung aufheben fehlgeschlagen	Das Git-Repository konnte die Verknüpfung mit dem Notebook nicht aufheben. Sie können erneut versuchen, die Verknüpfung aufzuheben.

Erstellen Sie ein neues Notebook mit einem zugehörigen Git-Repository in EMR Studio

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

So erstellen Sie ein Notebook und ordnen es Git-Repositorys in der alten Amazon-EMR-Konsole zu

1. Folgen Sie den Anweisungen unter [Erstellen Sie ein Notizbuch in EMR Studio](#).
2. Wählen Sie für Security group (Sicherheitsgruppe) die Option Use your own security group (Eigene Sicherheitsgruppe verwenden) aus.

Note

Die Sicherheitsgruppen für Ihr Notebook müssen eine Regel für ausgehenden Datenverkehr enthalten, sodass das Notebook Datenverkehr über den Cluster an das Internet weiterleiten kann. Wir empfehlen, eigene Sicherheitsgruppen zu erstellen. Weitere Informationen finden Sie unter [EC2 Sicherheitsgruppen für EMR Notebooks angeben](#).

3. Wählen Sie für Git repositories (Git-Repositorys) unter Choose repository (Repository wählen) das Repository aus, das dem Notebook zugeordnet werden soll.
 1. Wählen Sie ein Repository aus, das als Ressource in Ihrem Konto gespeichert ist, und wählen Sie dann Save (Speichern).
 2. Um ein neues Repository als Ressource in Ihrem Konto hinzuzufügen, wählen Sie add a new repository (Neues Repository hinzufügen). Führen Sie den Workflow Add repository (Repository hinzufügen) in einem neuen Fenster durch.

Verwenden Sie Git-Repositorys in einem EMR Studio-Notebook

Note

EMR Notebooks sind als EMR Studio-Workspaces in der Konsole verfügbar. Mit der Schaltfläche „Arbeitsbereich erstellen“ in der Konsole können Sie neue Notizbücher erstellen. Um auf Workspaces zuzugreifen oder diese zu erstellen, benötigen EMR-Notebook-Benutzer zusätzliche IAM-Rollenberechtigungen. Weitere Informationen finden Sie unter [Amazon EMR Notebooks sind Amazon EMR Studio Workspaces in der Konsole und Amazon EMR-Konsole](#).

Wenn Sie ein Notizbuch öffnen, können Sie wählen, ob Sie in Jupyter öffnen JupyterLab oder In Jupyter öffnen möchten.

Wenn Sie das Notebook in Jupyter öffnen, wird eine Liste erweiterbarer Dateien und Ordner im Notebook angezeigt. Sie können Git-Befehle wie die folgenden manuell in einer Notebook-Zelle ausführen.

```
!git pull origin primary
```

Zum Öffnen der zusätzlichen Repositorys wechseln Sie in andere Ordner.

Wenn Sie das Notizbuch mit einer JupyterLab Oberfläche öffnen möchten, können Sie die vorinstallierte JupyterLab Git-Erweiterung verwenden. Weitere Informationen über die Erweiterung finden Sie unter [jupyterlab-git](#).

Planung, Konfiguration und Start von Amazon EMR-Clustern

In diesem Abschnitt werden die Konfigurationsoptionen und Anweisungen für das Planen, Konfigurieren und Starten von Clustern mit Amazon EMR beschrieben. Bevor Sie einen Cluster starten, treffen Sie Entscheidungen hinsichtlich Ihres Systems auf der Grundlage der Daten, die Sie verarbeiten, und Ihrer Anforderungen an Kosten, Geschwindigkeit, Kapazität, Verfügbarkeit, Sicherheit und Verwaltbarkeit. Ihre Entscheidungen betreffen u. a. Folgendes:

- Welche Region in einem Cluster ausgeführt wird, wo und wie Daten gespeichert werden und wie Ergebnisse ausgegeben werden. Siehe [Amazon EMR-Clusterstandort und Datenspeicher konfigurieren](#).
- Wenn Sie Amazon-EMR-Cluster in Outposts oder Local Zones ausführen. Weitere Informationen unter [EMR-Cluster auf AWS Outposts](#) oder [EMR-Cluster in AWS Local Zones](#).
- Die Frage, ob ein Cluster von langer Dauer oder vorübergehend ist, und welche Software dort ausgeführt wird. Siehe [Konfiguration eines Amazon EMR-Clusters zum Fortfahren oder Beenden nach der Ausführung des Schritts](#) und [Konfigurieren Sie Anwendungen, wenn Sie Ihren Amazon EMR-Cluster starten](#).
- Wenn ein Cluster einen einzelnen Primärknoten oder drei Primärknoten besitzt. Siehe [Planung und Konfiguration primärer Knoten in Ihrem Amazon EMR-Cluster](#).
- Die Hardware- und Netzwerkoptionen, mit denen Kosten, Leistung und Verfügbarkeit Ihrer Anwendung optimiert werden. Siehe [Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren](#).
- Die Einrichtung von Clustern für eine leichtere Verwaltung sowie die Überwachung von Aktivitäten, Leistung und Zustand. Siehe [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#) und [Taggen und kategorisieren Sie Amazon EMR-Cluster-Ressourcen](#).
- Authentifizierung und Autorisierung des Zugriffs auf Cluster-Ressourcen und Verschlüsselung von Daten. Siehe [Sicherheit in Amazon EMR](#).
- Die Integration in andere Software und Services. Siehe [Integration von Treibern und Drittanbieteranwendungen auf Amazon EMR](#).

Schnelles Starten eines Amazon EMR-Clusters

Gehen Sie wie folgt vor, um in wenigen Minuten einen Amazon EMR-Cluster zu starten.

Um schnell einen Cluster mit der Konsole zu starten

1. Melden Sie sich bei <https://console.aws.amazon.com/emr/Clusters> an und öffnen Sie die Amazon EMR-Konsole. AWS Management Console
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Geben Sie auf der Seite Cluster erstellen Werte für die bereitgestellten Felder ein oder wählen Sie sie aus. Das persistente Übersichtsfenster zeigt eine Echtzeitanzeige Ihrer aktuell ausgewählten Clusteroptionen an. Wählen Sie im Übersichtsfenster eine Überschrift aus, um zum entsprechenden Abschnitt zu navigieren und Anpassungen vorzunehmen. Ihr Clustername darf die Zeichen <, >, \$, | oder ` (Backtick) nicht enthalten. Sie müssen alle erforderlichen Konfigurationen abschließen, bevor Sie Cluster erstellen auswählen können.
4. Wählen Sie Cluster erstellen aus, um die Konfiguration wie im Image gezeigt zu akzeptieren.
5. Dadurch wird die Cluster-Detailseite geöffnet. Suchen Sie den Cluster-Status neben dem Clusternamen. Der Status sollte sich während des Clustererstellungsprozesses von Startet zu Läuft zu Warten ändern. Möglicherweise müssen Sie das Aktualisierungssymbol oben rechts auswählen oder Ihren Browser aktualisieren, um Updates zu erhalten.

Wenn sich der Status in Warten ändert, ist Ihr Cluster betriebsbereit und bereit, Schritte und SSH-Verbindungen anzunehmen.

Amazon EMR-Clusterstandort und Datenspeicher konfigurieren

In diesem Abschnitt wird beschrieben, wie Sie die Region für einen Cluster konfigurieren, welche verschiedenen Dateisysteme mit Amazon EMR zur Verfügung stehen und wie sie verwendet werden. Außerdem wird erläutert, wie Sie Daten vorbereiten oder bei Bedarf in Amazon EMR hochladen und wie Sie einen Ausgabespeicherort für Protokolldateien und andere Ausgabedatendateien, die Sie konfigurieren, vorbereiten.

Themen

- [Wählen Sie eine AWS Region für Ihren Amazon EMR-Cluster](#)
- [Arbeiten mit Speicher- und Dateisystemen mit Amazon EMR](#)
- [Eingabedaten für die Verarbeitung mit Amazon EMR vorbereiten](#)
- [Einen Speicherort für die Amazon EMR-Clusterausgabe konfigurieren](#)

Wählen Sie eine AWS Region für Ihren Amazon EMR-Cluster

Amazon Web Services wird auf Servern in Rechenzentren auf der ganzen Welt ausgeführt. Die Rechenzentren sind nach geografischer Region organisiert. Wenn Sie einen Amazon-EMR-Cluster starten, müssen Sie eine Region angeben. So können Sie eine Region auswählen, die die Latenz oder die Kosten verringert oder behördliche Vorschriften erfüllt. Die Liste der von Amazon EMR unterstützten Regionen und Endpunkte finden Sie unter [Regionen und Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

Um eine optimale Leistung zu erzielen, sollten Sie den Cluster in derselben Region starten wie Ihre Daten. Wenn sich der Amazon-S3-Bucket, in dem Ihre Eingabedaten gespeichert sind, beispielsweise in der Region USA West (Oregon) befindet, sollten Sie Ihren Cluster in der Region USA West (Oregon) starten, um Gebühren für die regionsübergreifende Datenübertragung zu vermeiden. Wenn Sie einen Amazon-S3-Bucket für den Empfang der Cluster-Ausgabe verwenden, sollten Sie diesen Bucket ebenfalls in der Region USA West (Oregon) erstellen.

Wenn Sie dem Cluster ein EC2 Amazon-Schlüsselpaar zuordnen möchten (erforderlich für die Anmeldung am Master-Knoten über SSH), muss das key pair in derselben Region wie der Cluster erstellt werden. Auch die Sicherheitsgruppen, die Amazon EMR zum Verwalten des Clusters erstellt, müssen sich in der gleichen Region wie der Cluster befinden.

Wenn Sie sich AWS-Konto am oder nach dem 17. Mai 2017 für eine registriert haben, AWS Management Console ist die Standardregion, in der Sie auf eine Ressource zugreifen, USA Ost (Ohio) (us-east-2). Für ältere Konten ist die Standardregion entweder US West (Oregon) (us-west-2) oder USA Ost (Nord-Virginia) (us-east-1). Weitere Informationen finden Sie unter [-Regionen und Endpunkte](#).

Einige AWS Funktionen sind nur in begrenzten Regionen verfügbar. Beispielsweise sind Cluster-Compute-Instances nur in der Region USA Ost (Nord-Virginia) verfügbar und die Region Asien-Pazifik (Sydney) unterstützt nur Hadoop 1.0.3 und höher. Prüfen Sie bei der Auswahl einer Region, dass die Feature, die Sie verwenden möchten, in dieser Region unterstützt werden.

Um eine optimale Leistung zu erzielen, verwenden Sie dieselbe Region für alle AWS Ressourcen, die mit dem Cluster verwendet werden. In der folgenden Tabelle werden die Regionsnamen den Services zugeordnet. Die Liste der Amazon-EMR-Regionen finden Sie unter [AWS-Regionen -Regionen und Endpunkte](#) in der Allgemeine Amazon Web Services-Referenz.

Auswählen einer Region mithilfe der Konsole

Ihre Standardregion wird links neben Ihren Kontoinformationen in der Navigationsleiste angezeigt. Um die Region sowohl auf der neuen als auch auf der alten Konsole zu wechseln, wähle das Drop-down-Menü Region und wähle eine neue Option aus.

Geben Sie eine Region an mit AWS CLI

Geben Sie eine Standardregion in an, AWS CLI indem Sie entweder den `aws configure` Befehl oder die `AWS_DEFAULT_REGION` Umgebungsvariable verwenden. Weitere Informationen finden Sie unter [Konfiguration der AWS Region](#) im AWS Command Line Interface Benutzerhandbuch.

Eine Region mithilfe eines SDK oder der API auswählen

Um eine Region mit einem SDK auszuwählen, Sie Ihre Anwendung für die Verwendung des Endpunkts dieser Region. Wenn Sie eine Client-Anwendung mit einem AWS -SDK erstellen, können Sie den Client-Endpunkt ändern, indem Sie `setEndpoint` wie im folgenden Beispiel gezeigt aufrufen:

```
client.setEndpoint("elasticmapreduce.us-west-2.amazonaws.com");
```

Nachdem Ihre Anwendung durch die Festlegung des Endpunkts eine Region angegeben hat, können Sie die Availability Zone für die EC2 Instances Ihres Clusters festlegen. Availability Zones sind eindeutige geografische Standorte. Sie sollen vor Fehlern in anderen Availability Zones schützen und eine kostengünstige Netzwerkkonnektivität mit geringer Latenz zu anderen Availability Zones in der gleichen Region bereitstellen. Eine Region umfasst eine oder mehr Availability Zones. Um die Leistung zu optimieren und die Latenz zu verkürzen, sollten sich alle Ressourcen in derselben Availability Zone befinden wie der Cluster, der sie verwendet.

Arbeiten mit Speicher- und Dateisystemen mit Amazon EMR

Amazon EMR und Hadoop bieten eine Vielzahl von Dateisystemen an, die Sie bei der Verarbeitung von Cluster-Schritten verwenden können. Sie geben über das Präfix des URI an, welches Dateisystem Sie nutzen möchten. Beispielsweise referenziert `s3://amzn-s3-demo-bucket1/path` einen Amazon-S3-Bucket mittels EMRFS. In der folgenden Tabelle werden die verfügbaren Dateisysteme sowie Empfehlungen zu ihrer Verwendung aufgeführt.

Amazon EMR und Hadoop verwenden bei der Ausführung eines Clusters normalerweise zwei oder mehr der folgenden Dateisysteme. HDFS und EMRFS sind zwei der wichtigsten mit Amazon EMR verwendeten Dateisysteme.

Important

Ab Amazon EMR Version 5.22.0 verwendet Amazon EMR AWS Signature Version 4 ausschließlich zur Authentifizierung von Anfragen an Amazon S3. Frühere Amazon EMR-Versionen verwenden in einigen Fällen AWS Signature Version 2, sofern in den Versionshinweisen nicht angegeben ist, dass ausschließlich Signature Version 4 verwendet wird. Weitere Informationen finden Sie unter [Authentifizieren von Anfragen \(AWS Signature Version 4\)](#) und [Authentifizieren von Anfragen \(AWS Signature Version 2\)](#) im Amazon Simple Storage Service Developer Guide.

Dateisystem	Präfix	Beschreibung
HDFS	hdfs:// (oder ohne Präfix)	HDFS ist ein verteiltes, skalierbares und portierbares Dateisystem für Hadoop. Ein Vorteil von HDFS ist, dass die Daten für die Hadoop-Cluster-Knoten zur Verwaltung der Cluster und die Hadoop-Cluster-Knoten für die Verwaltung der einzelnen Schritte bekannt sind. Weitere Informationen finden Sie in der Hadoop-Dokumentation. HDFS wird von den Master- und Core-Knoten verwendet. Ein Vorteil ist, dass es schneller ist. Ein Nachteil ist, dass es ein flüchtiger Speicher ist. Dieser wird beim Beenden des Clusters verworfen. Es eignet sich am besten für die Zwischenspeicherung der Ergebnisse von zwischengeschalteten Auftragsverlaufsschritten.
EMRFS	s3://	EMRFS ist eine Implementierung des Hadoop-Dateisystems, die zum direkten Lesen und Schreiben regulärer Dateien aus Amazon EMR zu Amazon S3 verwendet wird. EMRFS bietet den Komfort, persistente Daten in Amazon S3 zur Verwendung mit Hadoop zu speichern und bietet gleichzeitig Funktionen wie serverseitige Amazon S3 S3-Verschlüsselung, read-after-write Konsistenz und Listenkonsistenz.

Dateisystem	Präfix	Beschreibung
		 Note Zuvor verwendete Amazon EMR die Dateisysteme s3n und s3a. Dies funktioniert zwar noch, doch wir empfehlen, dass Sie das URI-Schema s3 verwenden. Es bietet mehr Leistung, Sicherheit und Zuverlässigkeit.
Lokales Dateisystem		Das lokale Dateisystem bezieht sich auf einen lokal verbundenen Datenträger. Wenn ein Hadoop-Cluster erstellt wird, wird jeder Knoten aus einer EC2 Instance erstellt, die über einen vorkonfigurierten Block von vorinstalliertem Festplattenspeicher verfügt, der als Instance-Speicher bezeichnet wird. Daten auf Instance-Speicher-Volumes bleiben nur während der Lebensdauer der jeweiligen Instanz erhalten. EC2 Instance-Speicher-Volumen eignen sich perfekt für die Speicherung von temporärer Daten, die sich ständig ändern (z. B. Puffer, Caches, Arbeitsdaten und andere temporäre Inhalte). Weitere Informationen finden Sie unter EC2 Amazon-Instance-Speicher. Das lokale Dateisystem wird von HDFS verwendet, aber Python wird auch vom lokalen Dateisystem aus ausgeführt, und Sie können wählen, ob Sie zusätzliche Anwendungsdateien auf Instance-Speicher-Volumes speichern möchten.

Dateisystem	Präfix	Beschreibung
Amazon-S3-Block-Dateisystem (veraltet)	s3bfs://	Das Amazon-S3-Block-Dateisystem ist ein veraltetes Dateispeichersystem. Es wird ausdrücklich von der Verwendung dieses Systems abgeraten.  Important Wir empfehlen, dass Sie dieses Dateisystem nicht verwenden. Es kann dazu führen, dass der Cluster ausfällt. Möglicherweise ist es jedoch für ältere Anwendungen erforderlich.

Zugriff auf Dateisysteme

Sie geben über das Präfix des URI (Uniform Resource Identifier) für den Zugriff auf die Daten an, welches Dateisystem Sie nutzen möchten. Die folgenden Verfahren veranschaulichen den Zugriff auf verschiedene Dateisystemarten.

So greifen Sie auf ein lokales HDFS zu

- Geben Sie in `hdfs:///`-Präfix im URI an. Amazon EMR löst Pfade ohne Präfix im URI für das lokale HDFS auf. Zum Beispiel URIs würden die beiden folgenden Dateien auf denselben Speicherort in HDFS aufgelöst.

```
hdfs:///path-to-data
/path-to-data
```

So greifen Sie auf ein Remote-HDFS zu

- Fügen Sie, wie in den folgenden Beispielen dargestellt, die IP-Adresse des Master-Knotens zum URI hinzu.

```
hdfs://master-ip-address/path-to-data  
  
master-ip-address/path-to-data
```

So greifen Sie auf Amazon S3 zu

- Verwenden Sie den `s3://`-Präfix.

```
s3://bucket-name/path-to-file-in-bucket
```

So greifen Sie auf das Amazon-S3-Block-Dateisystem zu

- Verwenden Sie dieses Dateisystem nur für ältere Anwendungen, die das Amazon-S3-Block-Dateisystem benötigen. Für den Zugriff auf Daten mit diesem Dateisystem verwenden Sie das `s3bfs://`-Präfix im URI.

Das Amazon-S3-Block-Dateisystem ist ein veraltetes Dateisystem, das für Uploads zu Amazon S3 mit einer Größe von mehr als 5 GB verwendet wurde. Mit der mehrteiligen Upload-Funktion, die Amazon EMR über das AWS Java SDK bereitstellt, können Sie große Dateien in das native Amazon S3 S3-Dateisystem hochladen, und das Amazon S3 S3-Blockdateisystem ist veraltet. Weitere Informationen zum mehrteiligen Upload für EMR finden [Sie unter Mehrteiligen Upload für Amazon S3 konfigurieren](#). Weitere Informationen zu S3-Objektgrößen- und Teilgrößenbeschränkungen finden Sie unter [Amazon S3 S3-Beschränkungen für mehrteilige Uploads](#) im Amazon Simple Storage Service-Benutzerhandbuch.

 Warning

Da dieses veraltete Dateisystem zu Beschädigungen führen kann, sollten Sie es vermeiden. Verwenden Sie stattdessen EMRFS.

```
s3bfs://bucket-name/path-to-file-in-bucket
```


Eingabedaten für die Verarbeitung mit Amazon EMR vorbereiten

Die meisten Clustern laden Eingabedaten und verarbeitet diese anschließend. Zum Laden von Daten müssen diese sich an einem Speicherort befinden, auf den der Cluster zugreifen kann und der ein Format hat, das der Cluster verarbeiten kann. Das gängigste Szenario ist das zum Hochladen von Eingabedaten in Amazon S3. Amazon EMR bietet Tools, mit denen Ihr Cluster Daten aus Amazon S3 importieren oder lesen kann.

Das Standardeingabeformat in Hadoop sind Textdateien. Sie können Hadoop jedoch anpassen und Tools zum Importieren von Daten in anderen Formaten verwenden.

Themen

- [Arten von Eingabedaten, die Amazon EMR akzeptieren kann](#)
- [Verschiedene Möglichkeiten, Daten in Amazon EMR zu übertragen](#)

Arten von Eingabedaten, die Amazon EMR akzeptieren kann

Das Standardeingabeformat für einen Cluster sind Textdateien, bei denen jede Zeile durch ein Zeilenvorschubzeichen (`\n`) getrennt ist. Dies ist das am häufigsten verwendete Eingabeformat.

Wenn Ihre Eingabedaten in einem anderen Format geschrieben werden müssen als Standardtextdateien, können Sie die Hadoop-Benutzeroberfläche `InputFormat` verwenden, um andere Eingabetypen anzugeben. Sie können auch eine Unterklasse der `FileInputFormat`-Klasse für den Umgang mit benutzerdefinierten Datentypen verwenden. Weitere Informationen finden Sie unter <http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/InputFormat.html>.

Wenn Sie Hive verwenden, können Sie a serializer/deserializer (SerDe) verwenden, um Daten aus einem bestimmten Format in HDFS einzulesen. [Weitere Informationen finden Sie unter https://cwiki.apache.org/confluence/display/Hive/SerDe](https://cwiki.apache.org/confluence/display/Hive/SerDe).

Verschiedene Möglichkeiten, Daten in Amazon EMR zu übertragen

Amazon EMR bietet mehrere Möglichkeiten, um Daten auf einen Cluster zu laden. Die häufigste Methode besteht im Hochladen der Daten zu Amazon S3 und der Verwendung der integrierten Features von Amazon EMR, um die Daten in Ihren Cluster zu laden. Sie können auch das Hadoop-Feature `DistributedCache` für den verteilten Cache verwenden, um Dateien von einem verteilten Dateisystem in das lokale Dateisystem zu übertragen. Die von Amazon EMR bereitgestellte Hive-Implementierung (Hive-Version 0.7.1.1 und höher) enthält Funktionen, die Sie zum Importieren und

Exportieren von Daten zwischen DynamoDB und einen Amazon-EMR-Cluster verwenden können. Wenn Sie große Datenmengen On-Premises verarbeiten müssen, kann der AWS Direct Connect - Service nützlich sein.

Themen

- [Daten aus Amazon S3 uploaden](#)
- [Daten mit AWS DataSync hochladen](#)
- [Dateien mit verteiltem Cache mit Amazon EMR importieren](#)
- [Erkennung und Verarbeitung komprimierter Dateien mit Amazon EMR](#)
- [Importieren Sie DynamoDB-Daten mit Amazon EMR in Hive](#)
- [Connect zu Daten mit Amazon EMR AWS Direct Connect her](#)
- [Laden Sie große Datenmengen für Amazon EMR hoch mit AWS Snowball Edge](#)

Daten aus Amazon S3 uploaden

Informationen zum Hochladen von Objekten in Amazon S3 finden Sie unter [Ein Objekts zu Ihrem Bucket hinzufügen](#) im Benutzerhandbuch zu Amazon Simple Storage Service. Weitere Informationen zur Verwendung von Amazon S3 mit Hadoop finden Sie unter <http://wiki.apache.org/hadoop/AmazonS3>.

Themen

- [Erstellen und Konfigurieren eines Amazon S3-Buckets](#)
- [Konfigurieren von mehrteiligen Uploads für Amazon S3](#)
- [Bewährte Methoden](#)
- [Daten in Amazon S3 Express One Zone hochladen](#)

Erstellen und Konfigurieren eines Amazon S3-Buckets

Amazon EMR verwendet das AWS SDK für Java zusammen mit Amazon S3, um Eingabedaten, Protokolldateien und Ausgabedaten zu speichern. Amazon S3 bezeichnet diese Speicherorte als Buckets. Buckets haben in Übereinstimmung mit den Amazon-S3- und DNS-Anforderungen bestimmte Einschränkungen und Bedingungen. Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

In diesem Abschnitt erfahren Sie, wie Sie Amazon S3 verwenden AWS Management Console , um Berechtigungen für einen Amazon S3 S3-Bucket zu erstellen und anschließend festzulegen. Sie

können Berechtigungen für einen Amazon-S3-Bucket auch über die Amazon-S3-API oder die AWS CLI erstellen und festlegen. Sie können Curl auch zusammen mit einer Änderung verwenden, um die entsprechenden Authentifizierungsparameter für Amazon S3 zu übergeben.

Weitere Informationen finden Sie in den folgenden Ressourcen:

- Informationen zur Bucket-Erstellung mittels Konsole finden Sie unter [Erstellen eines Buckets](#) im Amazon-S3-Benutzerhandbuch.
- Informationen zum Erstellen und Arbeiten mit Buckets mithilfe von finden Sie unter [Verwenden von S3-Befehlen auf hoher Ebene mit dem AWS Command Line Interface](#) im Amazon S3 S3-Benutzerhandbuch. AWS CLI
- Informationen zum Erstellen eines Buckets mithilfe eines SDK finden Sie unter [Beispiele für die Erstellung eines Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.
- Informationen zum Arbeiten mit Buckets über Curl finden Sie unter [Amazon-S3-Authentifizierungstool für Curl](#).
- Weitere Informationen zum Angeben regionsspezifischer Buckets finden Sie unter [Zugreifen auf einen Bucket](#) im Benutzerhandbuch für Amazon Simple Storage Service.
- Informationen zum Arbeiten mit Buckets unter Verwendung von Amazon S3 Access Points finden Sie unter [Verwenden eines Alias im Bucket-Stil für Ihren Zugangspunkt](#) im Amazon-S3-Benutzerhandbuch. Sie können Amazon S3 Access Points problemlos mit dem Alias von Amazon S3 Access Points anstelle des Amazon-S3-Bucket-Namens verwenden. Sie können den Alias Amazon S3 Access Point sowohl für bestehende als auch für neue Anwendungen verwenden, darunter Spark, Hive, Presto und andere.

 Note

Wenn Sie für einen Bucket die Option „Protokollierung“ aktivieren, werden nur Bucket-Zugriffslogs aktiviert und nicht Amazon-EMR-Cluster-Logs.

Während der Bucket-Erstellung oder danach können Sie die entsprechenden Berechtigungen für den Zugriff auf den Bucket festlegen, abhängig von Ihrer Anwendung. Hierbei sollten Sie sich selbst (als Eigentümer) Lese- und Schreibzugriff und anderen autorisierten Benutzern Lesezugriff erteilen.

Erforderliche Amazon-S3-Buckets müssen vorhanden sein, bevor Sie einen Cluster erstellen können. Sie müssen alle erforderlichen Skripts und Daten auf Amazon S3 hochladen, auf die im Cluster

verwiesen wird. In der folgenden Tabelle werden Beispiele für Speicherorte für Daten, Skripts und Protokolldateien beschrieben.

Konfigurieren von mehrteiligen Uploads für Amazon S3

Amazon EMR unterstützt den mehrteiligen Amazon S3 S3-Upload über das AWS SDK for Java. Mit dem mehrteiligen Upload können Sie ein einzelnes Objekt in mehreren Teilen hochladen. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt.

Weitere Informationen finden Sie unter [Mehrteiliger Upload – Übersicht](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Darüber hinaus stellt Amazon EMR Eigenschaften bereit, mit denen Sie die Bereinigung fehlgeschlagener mehrteiliger Uploads genauer steuern können.

In der folgenden Tabelle werden die Amazon-EMR-Konfigurationsparameter für mehrteilige Uploads beschrieben. Sie können diese mit der Konfigurationsklassifizierung `core-site` konfigurieren. Weitere Informationen finden Sie unter [Konfigurieren von Anwendungen](#) in den Amazon-EMR-Versionshinweisen.

Name des Konfigurationsparameters	Standardwert	Beschreibung
<code>fs.s3n.multipart.uploads.enabled</code>	<code>true</code>	Dieser Boolesche Typ gibt an, ob mehrteilige Uploads aktiviert werden sollen. Wenn EMRFS konsistente Ansicht aktiviert ist, sind mehrteilige Uploads standardmäßig aktiviert. Eine Festlegung dieses Werts auf <code>false</code> wird ignoriert.
<code>fs.s3n.multipart.uploads.split.size</code>	134217728	Gibt die maximale Größe eines Teils in Byte an, bevor EMRFS einen neuen Teil-Upload startet, wenn die Funktion für mehrteilige Uploads aktiviert ist. Der Mindestwert ist 5242880 (5 MB). Wenn ein kleinerer Wert angegeben wird, wird 5242880 verwendet.

Name des Konfigurationsparameters	Standardwert	Beschreibung
		. Der Höchstwert ist 5368709120 (5 GB). Wenn ein größerer Wert angegeben wird, wird 5368709120 verwendet. Wenn die clientseitige EMRFS-Verschlüsselung und der Amazon S3 Optimized Committer deaktiviert sind, steuert dieser Wert auch die maximale Größe, die eine Datendatei erreichen kann, bis EMRFS zum Hochladen der Datei anstelle einer PutObject -Anfrage mehrteilige Uploads verwendet. Weitere Informationen finden Sie unter
fs.s3n.ssl.enabled	true	Dieser Boolesche Typ gibt an, ob HTTP oder HTTPS verwendet werden soll.
fs.s3.buckets.create.enabled	false	Ein boolescher Typ, der angibt, ob ein Bucket erstellt werden soll, wenn er nicht vorhanden ist. Wenn Sie dies auf false festlegen, wird eine Ausnahme für CreateBucket -Operationen ausgelöst.
fs.s3.multipart.clean.enabled	false	Ein boolescher Typ, der angibt, ob unvollständige mehrteilige Uploads regelmäßig im Hintergrund bereinigt werden sollen.
fs.s3.multipart.clean.age.threshold	604800	Ein long-Typ, der das Mindestalter eines mehrteiligen Uploads in Sekunden angibt, bevor er zur Bereinigung vorgesehen wird. Die Standardeinstellung ist eine Woche.

Name des Konfigurationsparameters	Standardwert	Beschreibung
<code>fs.s3.multipart.uploads.enabled.jitter.max</code>	10000	Eine integer-Typ, der den maximalen Betrag für zufällige Jitter-Verzögerungen in Sekunden angibt, die der festen Verzögerung von 15 Minuten hinzugefügt werden, bevor die nächste Bereinigung geplant wird.

So deaktivieren Sie mehrteilige Uploads

Console

Um mehrteilige Uploads mit der Konsole zu deaktivieren

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Geben Sie in Softwareeinstellungen bearbeiten die folgende Konfiguration ein:
`classification=core-site,properties=[fs.s3n.multipart.uploads.enabled=false]`.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um den mehrteiligen Upload zu deaktivieren, verwenden Sie AWS CLI

In diesem Verfahren wird erläutert, wie Sie mehrteilige Uploads mithilfe der AWS CLI deaktivieren. Um mehrteilige Uploads zu deaktivieren, geben Sie den Befehl `create-cluster` mit dem Parameter `--bootstrap-actions` ein.

1. Erstellen Sie eine Datei mit dem Namen `myConfig.json` und dem folgenden Inhalt und speichern Sie sie in dem Verzeichnis, in dem Sie den Befehl ausführen:

```
[
```

```
{
  "Classification": "core-site",
  "Properties": {
    "fs.s3n.multipart.uploads.enabled": "false"
  }
}
```

2. Geben Sie den folgenden Befehl ein und *myKey* ersetzen Sie ihn durch den Namen Ihres EC2 key pair.

 Note

Linux-Zeilenfortsetzungszeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.9.0 --applications Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --configurations file://myConfig.json
```

API

So deaktivieren Sie den mehrteiligen Upload mithilfe der API

- Informationen zur programmgesteuerten Verwendung von Amazon S3-Multipart-Uploads finden Sie unter [Verwenden des AWS SDK für Java für Multipart-Uploads](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Weitere Informationen zum AWS SDK for Java finden Sie unter [AWS SDK for Java](#).

Bewährte Methoden

Nachfolgend sind die Empfehlungen für die Nutzung von Amazon-S3-Buckets mit EMR-Clustern aufgeführt.

Aktivieren von Versioning

Versioning ist eine empfohlene Konfiguration für Ihre Amazon S3-Buckets. Durch das Aktivieren von Versioning stellen Sie sicher, dass Sie auch versehentlich gelöschte oder überschriebene Daten wiederhergestellt werden können. Weitere Informationen finden Sie unter [Verwenden von Versionsverwaltung](#) im Benutzerhandbuch für Amazon Simple Storage Service.

Bereinigung mehrteiliger Uploads fehlgeschlagen

EMR-Cluster-Komponenten verwenden standardmäßig mehrteilige Uploads über das AWS SDK for Java mit Amazon S3 APIs, um Protokolldateien zu schreiben und Daten in Amazon S3 auszugeben. Informationen zum Ändern von Eigenschaften im Zusammenhang mit dieser Konfiguration über Amazon EMR finden Sie unter [Konfigurieren von mehrteiligen Uploads für Amazon S3](#). Es kann vorkommen, dass das Hochladen einer großen Datei zu einem unvollständigen mehrteiligen Upload in Amazon S3 führt. Wenn ein mehrteiliger Upload nicht erfolgreich abgeschlossen werden kann, belegt der laufende Vorgang Ihren Bucket und es fallen Speichergebühren an. Wir empfehlen die folgenden Optionen, um eine übermäßige Dateispeicherung zu vermeiden:

- Verwenden Sie für mit Amazon EMR verwendete Buckets eine Lebenszyklus-Konfigurationsregel in Amazon S3, um unvollständige mehrteilige Uploads drei Tage nach dem Startdatum des betreffenden Uploads zu entfernen. Mit Lebenszyklus-Konfigurationsregeln können Sie Speicherklasse und Lebensdauer von Objekten steuern. Weitere Informationen finden Sie unter [Verwaltung des Objektlebenszyklus](#) und [Abbrechen unvollständiger mehrteiliger Uploads mit einer Bucket-Lebenszyklusrichtlinie](#).
- Sie aktivieren das Amazon-EMR-Feature für die Bereinigung mehrteiliger Uploads, indem Sie `fs.s3.multipart.clean.enabled` auf `true` festlegen und andere Bereinigungsparameter optimieren. Diese Funktion ist bei einem hohen Volumen, einem großem Umfang und Clustern mit begrenzter Betriebszeit nützlich. In diesem Fall ist der `DaysAfterInitiation`-Parameter einer Lebenszyklus-Konfigurationsregel möglicherweise zu lang, selbst wenn er auf das Minimum eingestellt ist, was zu Spitzen im Amazon-S3-Speicher führt. Die mehrteilige Bereinigung von Amazon EMR ermöglicht eine genauere Steuerung. Weitere Informationen finden Sie unter [Konfigurieren von mehrteiligen Uploads für Amazon S3](#).

Versionsmarkierungen verwalten

Sie sollten eine Lebenszyklus-Konfigurationsregel in Amazon S3 aktivieren, um abgelaufene Objektlöschmarkierungen für versionierte Buckets, die Sie mit Amazon EMR verwenden, zu entfernen. Beim Löschen eines Objekts in einem versionierten Bucket wird eine Löschmarkierung

erstellt. Wenn anschließend alle vorherigen Versionen des Objekts ablaufen, verbleibt eine Löschmarkierung für abgelaufene Objekte im Bucket. Löschmarkierungen werden Ihnen zwar nicht berechnet, die Entfernung abgelaufener Löschmarkierungen kann jedoch die Leistung von LIST-Anfragen verbessern. Weitere Informationen finden Sie unter [Lebenszykluskonfiguration für einen Bucket mit Versionsverwaltung](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Bewährte Methoden zur Leistungssteigerung

Je nach Workloads können bestimmte Nutzungsarten von EMR-Clustern und Anwendungen in diesen Clustern zu einer hohen Anzahl von Anfragen an einen Bucket führen. Weitere Informationen finden Sie unter [Erwägungen zur Anforderungsrate und Leistung](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Daten in Amazon S3 Express One Zone hochladen

Übersicht

Mit Amazon EMR 6.15.0 und höher können Sie Amazon EMR mit Apache Spark in Verbindung mit der Speicherklasse [Amazon S3 Express One Zone](#) verwenden, um die Leistung Ihrer Spark-Aufträge zu verbessern. Amazon EMR-Versionen 7.2.0 und höher unterstützen HBase auch Flink und Hive, sodass Sie auch von S3 Express One Zone profitieren können, wenn Sie diese Anwendungen verwenden. S3 Express One Zone ist eine S3-Speicherklasse für Anwendungen, die häufig mit Hunderttausenden Anfragen pro Sekunde auf Daten zugreifen. Zum Zeitpunkt seiner Veröffentlichung bietet S3 Express One Zone den Cloud-Objektspeicher mit der niedrigsten Latenz und der höchsten Leistung in Amazon S3.

Voraussetzungen

- Berechtigungen für S3 Express One Zone – Wenn S3 Express One Zone eine Aktion wie GET, LIST oder PUT für ein Amazon-S3-Objekt aufruft, ruft die Speicherklasse `CreateSession` in Ihrem Namen auf. Ihre IAM-Richtlinie muss die `s3express:CreateSession`-Genehmigung zulassen, damit der S3A-Konnektor die `CreateSession`-API aufrufen kann. Ein Beispielrichtlinie mit dieser Berechtigung finden Sie unter [Erste Schritte mit Amazon S3 Express One Zone](#).
- S3A-Konnektor – Um Ihren Spark-Cluster für den Zugriff auf Daten aus einem Amazon-S3-Bucket zu konfigurieren, der die Speicherklasse S3 Express One Zone verwendet, müssen Sie den Apache-Hadoop-Konnektor S3A verwenden. Um den Connector zu verwenden, stellen Sie sicher, dass alle S3 URIs das Schema verwenden. `s3a` Wenn dies nicht der Fall ist, können Sie die Dateisystemimplementierung, die Sie für `s3`- und `s3n`-Schemata verwenden, ändern.

Um das s3-Schema zu ändern, geben Sie die folgenden Clusterkonfigurationen an:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Um das s3n-Schema zu ändern, geben Sie die folgenden Clusterkonfigurationen an:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3n.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3n.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Erste Schritte mit Amazon S3 Express One Zone

Themen

- [Eine Berechtigungsrichtlinie erstellen](#)
- [Ihren Cluster erstellen und konfigurieren](#)
- [Konfigurationsübersicht](#)

Eine Berechtigungsrichtlinie erstellen

Bevor Sie einen Cluster erstellen können, der Amazon S3 Express One Zone verwendet, müssen Sie eine IAM-Richtlinie erstellen, die an das EC2 Amazon-Instance-Profil für den Cluster angehängt wird. Die Richtlinie muss über Berechtigungen für den Zugriff auf die Speicherklasse S3 Express One Zone verfügen. Die folgende Beispielrichtlinie zeigt, wie die erforderliche Berechtigung gewährt wird. Nachdem Sie die Richtlinie erstellt haben, ordnen Sie die Richtlinie der Instance-Profilrolle zu, mit der Sie Ihren EMR-Cluster erstellen, wie im Abschnitt [Ihren Cluster erstellen und konfigurieren](#) beschrieben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": "arn:aws:s3express:region-code:account-id:bucket/amzn-s3-demo-  
bucket",
      "Action": [
        "s3express:CreateSession"
      ]
    }
  ]
}
```

Ihren Cluster erstellen und konfigurieren

Erstellen Sie als Nächstes einen Cluster, auf dem Spark HBase, Flink oder Hive mit S3 Express One Zone ausgeführt wird. Die folgenden Schritte beschreiben einen allgemeinen Überblick über die Erstellung eines Clusters in der AWS Management Console:

1. Navigieren Sie zur Amazon-EMR-Konsole und wählen Sie in der Seitenleiste Cluster aus. Wählen Sie dann Create cluster (Cluster erstellen) aus.
2. Wenn Sie Spark verwenden, wählen Sie Amazon EMR-Version `emr-6.15.0` oder höher. Wenn Sie Flink oder Hive verwenden HBase, wählen Sie `emr-7.2.0` eine höhere Version.
3. Wählen Sie die Anwendungen aus, die Sie in Ihren Cluster aufnehmen möchten, z. B. Spark oder HBase Flink.
4. Um Amazon S3 Express One Zone zu aktivieren, geben Sie im Abschnitt Softwareeinstellungen eine Konfiguration ein, die dem folgenden Beispiel ähnelt. Die Konfigurationen und empfohlenen Werte werden in dem Abschnitt [Konfigurationsübersicht](#) beschrieben, der diesem Verfahren folgt.

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3a.aws.credentials.provider":
"software.amazon.awssdk.auth.credentials.InstanceProfileCredentialsProvider",
      "fs.s3a.change.detection.mode": "none",
      "fs.s3a.endpoint.region": "aa-example-1",
      "fs.s3a.select.enabled": "false"
    }
  }
]
```

```
    }
  },
  {
    "Classification": "spark-defaults",
    "Properties": {
      "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
    }
  }
]
```

5. Wählen Sie im Abschnitt EC2 Instanzprofil für Amazon EMR aus, ob Sie eine vorhandene Rolle und eine Rolle mit der angehängten Richtlinie verwenden möchten, die Sie im obigen [Eine Berechtigungsrichtlinie erstellen](#) Abschnitt erstellt haben.
6. Konfigurieren Sie die restlichen Cluster-Einstellungen entsprechend Ihrer Anwendung und wählen Sie dann Create cluster (Cluster erstellen) aus.

Konfigurationsübersicht

In den folgenden Tabellen werden die Konfigurationen und vorgeschlagenen Werte beschrieben, die Sie angeben sollten, wenn Sie einen Cluster einrichten, der S3 Express One Zone mit Amazon EMR verwendet, wie im Abschnitt [Ihren Cluster erstellen und konfigurieren](#) beschrieben.

S3A-Konfigurationen

Parameter	Standardwert	Empfohlener Wert	Erklärung
fs.s3a.aws.credentials.provider	Wenn nicht angegeben, wird AWSCredentialsProviderList in der folgenden Reihenfolge verwendet : TemporaryAWSCredentialsProvider , SimpleAWSCredentials	software.amazon.awssdk.auth.credentials.InstanceProfileCredentialsProvider	Die Amazon-EMR-Instance-Profilrolle sollte die Richtlinie haben, die es dem S3A-Dateisystem ermöglicht, s3express :CreateSession aufzurufen. Andere Anmeldeinformationen sind ebenfalls möglich,

Parameter	Standardwert	Empfohlener Wert	Erklärung
	<code>lsProvider ,EnvironmentVariablesProvider ,IAMInstanceCredentialsProvider</code> .		wenn sie über die Berechtigungen für S3 Express One Zone verfügen.
<code>fs.s3a.endpoint.region</code>	Null	Der AWS-Region Ort, an dem Sie den Bucket erstellt haben.	Die Logik zur Regionsauflösung funktioniert nicht mit der Speicherklasse S3 Express One Zone.
<code>fs.s3a.select.enabled</code>	<code>true</code>	<code>false</code>	Amazon S3 select wird mit der Speicherklasse S3 Express One Zone nicht unterstützt.
<code>fs.s3a.change.detection.mode</code>	<code>server</code>	Keine	Die Änderungserkennung von S3A erfolgt, indem MD5-basierte etags geprüft werden. Die Speicherklasse S3 Express One Zone unterstützt MD5 checksums nicht.

Spark-Konfigurationen

Parameter	Standardwert	Empfohlener Wert	Erklärung
<code>spark.sql.sources.fastS3PartitionDiscovery.enabled</code>	<code>true</code>	<code>false</code>	Die interne Optimierung verwendet einen S3-API-Parameter, den die Speicherklasse S3 Express One Zone nicht unterstützt.

Hive-Konfigurationen

Parameter	Standardwert	Empfohlener Wert	Erklärung
<code>hive.exec.fast.s3.partition.discovery.enabled</code>	<code>true</code>	<code>false</code>	Die interne Optimierung verwendet einen S3-API-Parameter, den die Speicherklasse S3 Express One Zone nicht unterstützt.

Überlegungen

Beachten Sie Folgendes, wenn Sie Apache Spark in Amazon EMR in die Speicherklasse S3 Express One Zone integrieren:

- Der S3A-Konnektor ist erforderlich, um S3 Express One Zone mit Amazon EMR zu verwenden. Nur S3A verfügt über die Features und Speicherklassen, die für die Interaktion mit S3 Express One Zone erforderlich sind. Schritte zum Einrichten des Konnektors finden Sie unter [the section called “Voraussetzungen”](#).
- Die Speicherklasse Amazon S3 Express One Zone wird nur mit Spark auf einem Amazon EMR-Cluster unterstützt, der auf Amazon EC2 läuft.
- Die Speicherklasse Amazon S3 Express One Zone unterstützt nur die SSE-S3-Verschlüsselung. Weitere Informationen finden Sie unter [Serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln \(SSE-S3\)](#).

- Die Speicherklasse Amazon S3 Express One Zone unterstützte keine Schreibvorgänge mit dem S3A FileOutputCommitter. Schreibvorgänge mit dem S3A FileOutputCommitter in Buckets von S3 Express One Zone führen zu einem Fehler: InvalidStorageClass: The storage class you specified is not valid.
- Amazon S3 Express One Zone wird mit Amazon EMR-Versionen 6.15.0 und höher auf EMR on EC2 Darüber hinaus wird es auf Amazon EMR-Versionen 7.2.0 und höher, auf Amazon EMR auf EKS und auf Amazon EMR Serverless unterstützt.

Daten mit AWS DataSync hochladen

AWS DataSync ist ein Online-Datenübertragungsservice, der den Prozess der Übertragung von Daten zwischen Ihrem lokalen Speicher und Speicherdiensten oder zwischen AWS Speicherdiensten vereinfacht, automatisiert und beschleunigt. AWS DataSync unterstützt eine Vielzahl von lokalen Speichersystemen wie Hadoop Distributed File System (HDFS), NAS-Dateiserver und selbstverwalteten Objektspeicher.

Der gängigste Weg, Daten auf einen Cluster zu übertragen, besteht darin, die Daten auf Amazon S3 hochzuladen und die integrierten Features von Amazon EMR zu verwenden, um die Daten auf Ihren Cluster zu laden.

DataSync kann Ihnen helfen, die folgenden Aufgaben zu erledigen:

- Replizieren Sie HDFS auf Ihrem Hadoop-Cluster auf Amazon S3 für Geschäftskontinuität
- HDFS nach Amazon S3 kopieren, um Ihre Data Lakes zu füllen
- Daten zwischen dem HDFS Ihres Hadoop-Clusters und Amazon S3 zur Analyse und Verarbeitung übertragen

Um Daten in Ihren S3-Bucket hochzuladen, setzen Sie zunächst einen oder mehrere DataSync Agenten im selben Netzwerk ein, in dem sich Ihr lokaler Speicher befindet. Ein Agent ist eine virtuelle Maschine (VM), die zum Lesen von Daten oder zum Schreiben von Daten an einem selbstverwalteten Speicherort verwendet wird. Anschließend aktivieren Sie Ihre Agenten in dem AWS-Konto und AWS-Region wo sich Ihr S3-Bucket befindet.

Nachdem Ihr Agent aktiviert wurde, erstellen Sie einen Quellstandort für Ihren On-Premises-Speicher, einen Zielort für Ihren S3-Bucket und eine Aufgabe. Eine Aufgabe ist ein Satz von zwei Speicherorten (Quelle und Ziel) und eine Reihe von Standardoptionen, die Sie verwenden, um das Verhalten der Aufgabe zu steuern.

Schließlich führen Sie Ihre DataSync Aufgabe aus, um Daten von der Quelle zum Ziel zu übertragen.

Weitere Informationen finden Sie unter [Erste Schritte mit AWS DataSync](#).

Dateien mit verteiltem Cache mit Amazon EMR importieren

DistributedCache ist ein Hadoop-Feature, das die Effizienz erhöhen kann, wenn eine Zuordnungs- oder Reduzierungs-Aufgabe Zugriff auf allgemeine Daten benötigt. Wenn Ihr Cluster von vorhandenen Anwendungen oder Binärdateien abhängt, die bei der Erstellung des Clusters nicht installiert sind, können Sie den DistributedCache zum Importieren dieser Dateien verwenden. Mit dieser Funktion kann ein Cluster-Knoten die importierten Dateien aus seinem lokalen Dateisystem lesen, anstatt die Dateien von anderen Cluster-Knoten abzurufen.

Weitere Informationen finden Sie unter <http://hadoop.apache.org/docs/stable/api/org/apache/hadoop/filecache/DistributedCache.html>.

Sie rufen den DistributedCache beim Erstellen des Clusters auf. Die Dateien werden vor dem Starten des Hadoop-Auftrags nur für die Dauer des Auftrags im Cache zwischengespeichert. Sie können Dateien, die in einem beliebigen Hadoop-kompatiblen Dateisystem wie HDFS oder Amazon S3 gespeichert sind, im Cache zwischenspeichern. Die Standardgröße des Datei-Caches ist 10 GB. Zum Ändern der Größe des Caches konfigurieren Sie den Hadoop-Parameter `local.cache.size` mithilfe der Bootstrap-Aktion neu. Weitere Informationen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#).

Themen

- [Unterstützte Dateitypen](#)
- [Speicherort der zwischengespeicherten Dateien](#)
- [Auf zwischengespeicherte Dateien über Streaming-Anwendungen zugreifen](#)
- [Auf zwischengespeicherte Dateien über Streaming-Anwendungen zugreifen](#)

Unterstützte Dateitypen

Der DistributedCache lässt sowohl einzelne Dateien als auch Archive zu. Einzelne Dateien werden schreibgeschützt zwischengespeichert. Für ausführbare und Binärdateien werden Ausführungsberechtigungen festgelegt.

Archive sind eine oder mehrere Dateien, die mit einem Hilfsprogramm verpackt wurden, z. B. `gzip`. DistributedCache übergibt die komprimierten Dateien an jeden Core-Knoten und dekomprimiert

das Archiv im Rahmen der Zwischenspeicherung. DistributedCacheunterstützt die folgenden Komprimierungsformate:

- zip
- tgz
- tar.gz
- tar
- jar

Speicherort der zwischengespeicherten Dateien

DistributedCache kopiert Dateien nur auf Core-Knoten. Wenn es im Cluster keine Core-Knoten gibt, kopiert der DistributedCache die Dateien zum Primärknoten.

Der DistributedCache weist die Cache-Dateien dem aktuellen Arbeitsverzeichnis des Mappers und Reducers mithilfe von symbolischen Links zu. Ein symbolischer Link (symlink) ist ein Alias für einen Dateispeicherort, nicht der tatsächliche Speicherort. Der Wert des Parameters, `yarn.nodemanager.local-dirs` in `yarn-site.xml`, gibt den Speicherort der temporären Dateien an. Amazon EMR legt diesen Parameter auf `/mnt/mapred` oder eine Variante basierend auf dem Instance-Typ und der EMR-Version fest. Eine Einstellung kann die Werte `/mnt/mapred` und `/mnt1/mapred` haben, da der Instance-Typ über zwei flüchtige Volumes verfügt. Cache-Dateien befinden sich in einem Unterverzeichnis des Speicherorts für temporäre Dateien unter `/mnt/mapred/taskTracker/archive`.

Wenn Sie eine einzelne Datei zwischenspeichern wird sie über den DistributedCache im Verzeichnis `archive` abgelegt. Wenn Sie ein Archiv zwischenspeichern, wird sie vom DistributedCache dekomprimiert und es wird im `/archive` ein Unterverzeichnis mit demselben Namen wie dem Archivdateinamen erstellt. Die einzelnen Dateien befinden sich im neuen Unterverzeichnis.

Sie können den DistributedCache nur bei Verwendung von Streaming verwenden.

Auf zwischengespeicherte Dateien über Streaming-Anwendungen zugreifen

Um aus Ihren Mapper- oder Reducer-Anwendungen auf die zwischengespeicherten Dateien zugreifen zu können, müssen Sie Ihrem Anwendungspfad das aktuelle Arbeitsverzeichnis (`./`) hinzufügen und die zwischengespeicherten Dateien so referenzieren, als würden sie sich im aktuellen Arbeitsverzeichnis befinden.

Auf zwischengespeicherte Dateien über Streaming-Anwendungen zugreifen

Sie können die AWS Management Console und die verwenden, um Cluster AWS CLI zu erstellen, die den verteilten Cache verwenden.

Console

So geben Sie verteilte Cache-Dateien mithilfe der neuen Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Schritte die Option Schritt hinzufügen aus. Dadurch wird das Dialogfeld Schritt hinzufügen geöffnet. Geben Sie im Feld Argumente die Dateien und Archive an, die im Cache gespeichert werden sollen. Die Größe der Datei (oder Gesamtgröße der Dateien in einer Archivdatei) muss geringer sein als die zugewiesene Cachegröße.

Wenn Sie eine einzelne Datei zum verteilten Cache hinzufügen möchten, geben Sie `-cacheFile` an, gefolgt vom Namen und Speicherort der Datei, dem Rautenzeichen (#) und dem Namen, den Sie der Datei geben möchten, wenn sie im lokalen Cache abgelegt wird. Im folgenden Beispiel wird gezeigt, wie eine einzelne Datei zum verteilten Cache hinzugefügt wird.

```
-cacheFile \  
s3://amzn-s3-demo-bucket/file-name#cache-file-name
```

Geben Sie `-cacheArchive` gefolgt von dem Speicherort der Dateien in Amazon S3, dem Rautenzeichen (#) und dann dem Namen ein, den Sie der Sammlung von Dateien im verteilten Cache geben möchten. Im folgenden Beispiel wird gezeigt, wie eine einzelne Datei zum verteilten Cache hinzugefügt wird.

```
-cacheArchive \  
s3://amzn-s3-demo-bucket/archive-name#cache-archive-name
```

Geben Sie die entsprechenden Werte in die anderen Dialogfelder ein. Die Optionen unterscheiden sich je nach Schritttyp. Um Ihren Schritt hinzuzufügen und das Dialogfeld zu verlassen, wählen Sie Schritt hinzufügen.

4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um verteilte Cache-Dateien mit dem zu spezifizieren AWS CLI

- Um einen Streaming-Schritt beim Erstellen eines Clusters zu senden, geben Sie den Befehl `create-cluster` mit dem Parameter `--steps` ein. Um verteilte Cache-Dateien mithilfe von anzugeben AWS CLI, geben Sie beim Senden eines Streaming-Schritts die entsprechenden Argumente an.

Wenn Sie eine einzelne Datei zum verteilten Cache hinzufügen möchten, geben Sie `-cacheFile` an, gefolgt vom Namen und Speicherort der Datei, dem Rautenzeichen (`#`) und dem Namen, den Sie der Datei geben möchten, wenn sie im lokalen Cache abgelegt wird.

Geben Sie `-cacheArchive` gefolgt von dem Speicherort der Dateien in Amazon S3, dem Rautenzeichen (`#`) und dann dem Namen ein, den Sie der Sammlung von Dateien im verteilten Cache geben möchten. Im folgenden Beispiel wird gezeigt, wie eine einzelne Datei zum verteilten Cache hinzugefügt wird.

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Example 1

Geben Sie den folgenden Befehl zum Starten eines Clusters und zum Senden eines Streaming-Schritts ein, der `-cacheFile` zum Hinzufügen einer Datei, `sample_dataset_cached.dat`, zum Cache verwendet.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
--instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py
s3://my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py","-
input","s3://my_bucket/my_input","-output","s3://my_bucket/my_output", "-
cacheFile","s3://my_bucket/sample_dataset.dat#sample_dataset_cached.dat"]
```

Wenn Sie die Instance-Anzahl ohne den `--instance-groups`-Parameter angeben, wird ein einzelner Primärknoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

Wenn Sie noch nicht die standardmäßige EMR-Dienstrolle und das EC2 Instanzprofil erstellt haben, geben Sie ein, um sie `aws emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl eingeben.

Example 2

Der folgende Befehl erstellt einen Streaming-Cluster und verwendet `-cacheArchive`, um dem Cache ein Dateiarchiv hinzuzufügen.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
--instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
program",ActionOnFailure=CONTINUE,Args=["--files", "s3://my_bucket/my_mapper.py
s3://my_bucket/my_reducer.py", "-mapper", "my_mapper.py", "-reducer", "my_reducer.py", "-
input", "s3://my_bucket/my_input", "-output", "s3://my_bucket/my_output", "-
cacheArchive", "s3://my_bucket/sample_dataset.tgz#sample_dataset_cached"]
```

Wenn Sie die Instance-Anzahl ohne den `--instance-groups`-Parameter angeben, wird ein einzelner Primärknoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

Wenn Sie noch nicht die standardmäßige EMR-Dienstrolle und das EC2 Instanzprofil erstellt haben, geben Sie ein, um sie `aws emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl eingeben.

Erkennung und Verarbeitung komprimierter Dateien mit Amazon EMR

Hadoop überprüft die Dateierweiterung zur Erkennung von komprimierten Dateien. Folgende Komprimierungstypen werden von Hadoop unterstützt: gzip, bzip2 und LZO. Sie müssen keine zusätzlichen Schritte ausführen, um Dateien dieser Komprimierungstypen zu extrahieren, da Hadoop diesen Vorgang für Sie erledigt.

[Um LZO-Dateien zu indizieren, können Sie die Hadoop-LZO-Bibliothek verwenden, die von hadoop-lzo heruntergeladen werden kann. https://github.com/kevinweil/](https://github.com/kevinweil/) Beachten Sie, dass es sich um eine Drittanbieter-Bibliothek handelt. Amazon EMR bietet daher keinen Entwickler-Support bei Verwendung dieses Tools. Informationen zur Nutzung finden Sie in der [Readme-Datei für hadoop-lzo](#).

Importieren Sie DynamoDB-Daten mit Amazon EMR in Hive

Die von Amazon EMR bereitgestellte Hive-Implementierung enthält Funktionalität, die Sie zum Importieren und Exportieren von Daten zwischen DynamoDB und einem Amazon-EMR-Cluster verwenden können. Dies ist nützlich, wenn Ihre Eingabedaten in DynamoDB gespeichert sind. Weitere Informationen finden Sie unter [Exportieren, Importieren, Abfragen und Verknüpfen von Tabellen in DynamoDB mit Amazon EMR](#).

Connect zu Daten mit Amazon EMR AWS Direct Connect her

AWS Direct Connect ist ein Service, mit dem Sie von Ihrem Rechenzentrum, Büro oder Ihrer Colocation-Umgebung aus eine private, dedizierte Netzwerkverbindung zu Amazon Web Services herstellen können. Wenn Sie über große Mengen an Eingabedaten verfügen, AWS Direct Connect kann die Verwendung Ihrer Netzwerkkosten senken, den Bandbreitendurchsatz erhöhen und ein einheitlicheres Netzwerkerlebnis bieten als internetbasierte Verbindungen. Weitere Informationen finden Sie im [AWS Direct Connect -Benutzerhandbuch](#).

Laden Sie große Datenmengen für Amazon EMR hoch mit AWS Snowball Edge

AWS Snowball Edge ist ein Service, mit dem Sie große Datenmengen schnell zwischen Amazon Simple Storage Service (Amazon S3) und Ihrem Datenspeicherort vor Ort übertragen faster-than-internet können. Snowball Edge unterstützt zwei Auftragstypen: Importaufträge und Exportaufträge. Importaufträge beinhalten eine Datenübertragung von einer On-Premises-Quelle zu einem Amazon-S3-Bucket. Exportaufträge beinhalten eine Datenübertragung aus einem Amazon-S3-Bucket zu einer On-Premises-Quelle. Bei beiden Auftragstypen sichern und schützen Snowball Edge-Geräte Ihre Daten, während regionale Spediteure sie zwischen Amazon S3 und Ihrem Datenspeicherort vor Ort transportieren. Snowball Edge-Geräte sind physisch robust und werden durch die AWS Key Management Service (AWS KMS) geschützt. Weitere Informationen finden Sie im [AWS Snowball Edge -Edge-Entwicklerhandbuch](#).

Einen Speicherort für die Amazon EMR-Clusterausgabe konfigurieren

Das häufigste Ausgabeformat eines Amazon-EMR-Clusters sind Textdateien, und zwar entweder komprimiert oder nicht komprimiert. Diese Dateien werden in der Regel in einen Amazon-S3-Bucket geschrieben. Dieser Bucket muss erstellt werden, bevor Sie den Cluster starten. Sie geben den S3-Bucket als Ausgabespeicherort an, wenn Sie den Cluster starten.

Weitere Informationen finden Sie unter den folgenden Themen:

Themen

- [Erstellen und Konfigurieren eines Amazon S3-Buckets](#)
- [Welche Formate kann Amazon EMR zurückgeben?](#)
- [So schreiben Sie mit Amazon EMR Daten in einen Amazon S3 S3-Bucket, den Sie nicht besitzen](#)
- [Möglichkeiten, die Ausgabe Ihres Amazon EMR-Clusters zu komprimieren](#)

Erstellen und Konfigurieren eines Amazon S3-Buckets

Amazon EMR (Amazon EMR) verwendet Amazon S3 zum Speichern von Eingabedaten, Protokolldateien und Ausgabedaten. Amazon S3 bezeichnet diese Speicherorte als Buckets. Buckets haben in Übereinstimmung mit den Amazon-S3- und DNS-Anforderungen bestimmte Einschränkungen und Bedingungen. Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Amazon Simple Storage Service-Entwicklerhandbuch.

Um einen Amazon-S3-Bucket zu erstellen, befolgen Sie die Anweisungen auf der Seite [Bucket erstellen](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Note

Wenn Sie im Assistenten Bucket erstellen die Protokollierung aktivieren, werden nur Bucket-Zugriffsprotokolle aktiviert und nicht Cluster-Protokolle.

Note

Weitere Informationen zur Angabe regionsspezifischer Buckets finden Sie unter [Buckets and Regions](#) im Amazon Simple Storage Service Developer Guide und [Available Region Endpoints for the AWS SDKs](#)

Nachdem Sie Ihren Bucket erstellt haben, können Sie die entsprechenden Zugriffsberechtigungen hierzu einrichten. Hierbei sollten Sie sich selbst (als Eigentümer) Lese- und Schreibzugriff erteilen. Wir empfehlen Ihnen dringend, bei der Konfiguration Ihres Buckets die [bewährten Sicherheitsmethoden für Amazon S3](#) zu befolgen.

Erforderliche Amazon-S3-Buckets müssen vorhanden sein, bevor Sie einen Cluster erstellen können. Sie müssen alle erforderlichen Skripts und Daten auf Amazon S3 hochladen, auf die im Cluster verwiesen wird. In der folgenden Tabelle werden Beispiele für Speicherorte für Daten, Skripts und Protokolldateien beschrieben.

Informationen	Beispielspeicherort auf Amazon S3
Skript oder Programm	s3://amzn-s3-demo-bucket1/script/MapScript.py
Protokolldateien	s3://amzn-s3-demo-bucket1/logs
Eingabedaten	s3://amzn-s3-demo-bucket1/input
Ausgabedaten	s3://amzn-s3-demo-bucket1/output

Welche Formate kann Amazon EMR zurückgeben?

Das Standardausgabeformat für einen Cluster ist Text mit Schlüssel-Wert-Paaren, die in einzelne Zeilen der Textdateien geschrieben werden. Dies ist das am häufigsten verwendete Ausgabeformat.

Wenn Ihre Ausgabedaten in einem anderen Format geschrieben werden müssen als Standardtextdateien, können Sie die Hadoop-Benutzeroberfläche `OutputFormat` verwenden, um andere Ausgabetypen anzugeben. Sie können auch eine Unterklasse der `FileOutputFormat`-Klasse für den Umgang mit benutzerdefinierten Datentypen verwenden. Weitere Informationen finden Sie unter <http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/OutputFormat.html>.

Wenn Sie einen Hive-Cluster starten, können Sie a serializer/deserializer (SerDe) verwenden, um Daten aus HDFS in ein bestimmtes Format auszugeben. [Weitere Informationen finden Sie unter https://cwiki.apache.org/confluence/display/Hive/SerDe](https://cwiki.apache.org/confluence/display/Hive/SerDe).

So schreiben Sie mit Amazon EMR Daten in einen Amazon S3 S3-Bucket, den Sie nicht besitzen

Wenn Sie eine Datei in einen Amazon Simple Storage Service (Amazon S3)-Bucket schreiben, können standardmäßig nur Sie die Datei lesen. Es wird davon ausgegangen, dass Sie Dateien in Ihre Buckets schreiben. Diese Standardeinstellung dient dem Schutz Ihrer Dateien.

Wenn Sie jedoch einen Cluster ausführen und möchten, dass die Ausgabe in den Amazon S3 S3-Bucket eines anderen AWS Benutzers schreibt und Sie möchten, dass dieser andere AWS Benutzer diese Ausgabe lesen kann, müssen Sie zwei Dinge tun:

- Bitten Sie den anderen AWS Benutzer, Ihnen Schreibberechtigungen für seinen Amazon S3 S3-Bucket zu gewähren. Der Cluster, den Sie starten, wird unter Ihren AWS Anmeldeinformationen

ausgeführt, sodass alle Cluster, die Sie starten, auch in den Bucket dieses anderen AWS Benutzers schreiben können.

- Legen Sie Leseberechtigungen für den anderen AWS Benutzer für die Dateien fest, die Sie oder der Cluster in den Amazon S3 S3-Bucket schreiben. Die einfachste Methode, diese Leseberechtigungen festzulegen, besteht darin, vorgefertigte Zugriffskontrolllisten (ACLs) zu verwenden, eine Reihe von vordefinierten Zugriffsrichtlinien, die von Amazon S3 definiert wurden.

Informationen darüber, wie der andere AWS Benutzer Ihnen Berechtigungen zum Schreiben von Dateien in den Amazon S3 S3-Bucket des anderen Benutzers gewähren kann, finden Sie unter [Bearbeiten von Bucket-Berechtigungen](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Damit Ihr Cluster ACLs beim Schreiben von Dateien in Amazon S3 die Funktion „Gespeichert“ verwenden soll, setzen Sie die `fs.s3.canned.acl` Cluster-Konfigurationsoption auf die zu verwendende ACL mit gespeicherten Daten. In der folgenden Tabelle sind die aktuell definierten Canned aufgeführt ACLs.

Vordefinierte ACL	Beschreibung
AuthenticatedRead	Gibt an, dass dem Eigentümer <code>Permission.FullControl</code> und dem Berechtigungsempfänger der Gruppe <code>GroupGrantee.AuthenticatedUsers</code> der Zugriff <code>Permission.Read</code> gewährt wird.
BucketOwnerFullControl	Gibt an, dass dem Bucket-Eigentümer <code>Permission.FullControl</code> gewährt wird. Der Bucket-Eigentümer muss nicht unbedingt derselbe wie der Objekteigentümer sein.
BucketOwnerRead	Gibt an, dass dem Bucket-Eigentümer <code>Permission.Read</code> gewährt wird. Der Bucket-Eigentümer muss nicht unbedingt derselbe wie der Objekteigentümer sein.
LogDeliveryWrite	Gibt an, dass dem Eigentümer <code>Permission.FullControl</code> und dem Berechtigungsempfänger der Gruppe <code>GroupGrantee.LogDelivery</code> der Zugriff <code>Permission.Write</code> gewährt wird, damit Zugriffsp rotokolle bereitgestellt werden können.

Vordefinierte ACL	Beschreibung
Private	Gibt an, dass dem Eigentümer <code>Permission.FullControl</code> gewährt wird.
PublicRead	Gibt an, dass dem Eigentümer <code>Permission.FullControl</code> und dem Berechtigungsempfänger der Gruppe <code>GroupGrantee.AllUsers</code> der Zugriff <code>Permission.Read</code> gewährt wird.
PublicReadWrite	Gibt an, dass dem Eigentümer <code>Permission.FullControl</code> und dem Berechtigungsempfänger der Gruppe <code>GroupGrantee.AllUsers</code> die Zugriffsberechtigungen <code>Permission.Read</code> und <code>Permission.Write</code> gewährt wird.

Die Cluster-Konfigurationsoptionen können auf vielfältige Weise festgelegt werden, je nach Typ des ausgeführten Clusters. Die folgenden Verfahren zeigen die Festlegung der Option für allgemeine Anwendungsfälle.

So schreiben Sie Dateien mithilfe von canned ACLs in Hive

- Legen Sie in der Hive-Eingabeaufforderung die Konfigurationsoption `fs.s3.canned.acl` auf die vordefinierte ACL fest, die der Cluster für zu Amazon S3 geschriebene Dateien festlegen soll. Stellen Sie für den Zugriff auf die Hive-Eingabeaufforderung eine Verbindung mit dem Master-Knoten über SSH her und geben Sie an der Hadoop-Eingabeaufforderung "Hive" ein. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Im folgenden Beispiel wird die Konfigurationsoption `fs.s3.canned.acl` auf `BucketOwnerFullControl` festgelegt. Dadurch erhält der Eigentümer des Amazon-S3-Buckets vollständige Kontrolle über die Datei. Beachten Sie: Der Festlegungsbefehl erfordert die Beachtung der Groß- und Kleinschreibung und enthält keine Anführungszeichen oder Leerzeichen.

```
hive> set fs.s3.canned.acl=BucketOwnerFullControl;
create table acl (n int) location 's3://amzn-s3-demo-bucket/acl/';
```

```
insert overwrite table acl select count(*) from acl;
```

Die beiden letzten Zeilen des Beispiels erstellen eine Tabelle, die in Amazon S3 gespeichert wird, und schreiben Daten in die Tabelle.

Um Dateien mit Canned ACLs in Pig zu schreiben

- Legen Sie in der Pig-Eingabeaufforderung die Konfigurationsoption `fs.s3.canned.acl` auf die vordefinierte ACL fest, die der Cluster für zu Amazon S3 geschriebene Dateien festlegen soll. Stellen Sie für den Zugriff auf die Pig-Eingabeaufforderung eine Verbindung mit dem Master-Knoten über SSH her und geben Sie an der Hadoop-Eingabeaufforderung "Pig" ein. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Im folgenden Beispiel wird die `fs.s3.canned.acl` Konfigurationsoption auf `BucketOwnerFullControl` gesetzt, wodurch der Besitzer des Amazon S3 S3-Buckets die vollständige Kontrolle über die Datei hat. Beachten Sie: Der Festlegungsbefehl enthält ein Leerzeichen vor dem Namen der vordefinierten ACL und keine Anführungszeichen.

```
pig> set fs.s3.canned.acl BucketOwnerFullControl;  
store some data into 's3://amzn-s3-demo-bucket/pig/acl';
```

Um Dateien mithilfe von Canned ACLs in einem benutzerdefinierten JAR zu schreiben

- Legen Sie mit Hadoop die Konfigurationsoption `fs.s3.canned.acl` mit `-D`-Flag fest. Das wird im Beispiel unten veranschaulicht.

```
hadoop jar hadoop-examples.jar wordcount  
-Dfs.s3.canned.acl=BucketOwnerFullControl s3://amzn-s3-demo-bucket/input s3://amzn-s3-demo-bucket/output
```

Möglichkeiten, die Ausgabe Ihres Amazon EMR-Clusters zu komprimieren

Es gibt verschiedene Möglichkeiten, die Ausgabe zu komprimieren, die aus der Datenverarbeitung resultiert. Welche Komprimierungstools Sie verwenden, hängt von den Eigenschaften Ihrer Daten ab. Durch die Komprimierung kann die Leistung verbessert werden, wenn Sie große Datenmengen übertragen.

Kompression der Ausgabedaten

Dies komprimiert die Ausgabe Ihres Hadoop-Auftrags. Wenn Sie `TextOutputFormat` das Ergebnis verwenden, handelt es sich um eine GZIP-komprimierte Textdatei. Wenn Sie schreiben, `SequenceFiles` ist das Ergebnis eine `SequenceFile`, die intern komprimiert ist. Dies kann aktiviert werden, indem Sie die Konfigurationseinstellung `"mapred.output.compress"` auf `"true"` setzen.

Wenn Sie einen Streaming-Auftrag ausführen, können Sie dies aktivieren, indem Sie dem Streaming-Auftrag diese Argumente übergeben.

```
-jobconf mapred.output.compress=true
```

Sie können auch mit einer Bootstrap-Aktion alle Auftragsausgaben automatisch komprimieren. So können Sie dies mit dem Ruby-Client ausführen:

```
--bootstrap-actions s3://elasticmapreduce/bootstrap-actions/configure-hadoop \  
--args "-s,mapred.output.compress=true"
```

Wenn Sie eine Custom Jar schreiben, können Sie die Ausgabekompression mit folgender Zeile bei der Erstellung Ihres Auftrags aktivieren:

```
FileOutputFormat.setCompressOutput(conf, true);
```

Intermediäre Datenkompression

Wenn Ihr Auftrag eine erhebliche Menge Daten von den Mappern zu den Reducern verlagert, können Sie eine Leistungsverbesserung durch Aktivierung der intermediären Kompression feststellen. Sie

komprimieren die Zuweisungsausgabe und dekomprimieren sie, wenn sie auf dem Core-Knoten eingeht. Die Konfigurationseinstellung ist "mapred.compress.map.output". Sie können sie ähnlich wie die Ausgabekompression aktivieren.

Wenn Sie eine Custom Jar schreiben, verwenden Sie den folgenden Befehl:

```
conf.setCompressMapOutput(true);
```

Verwenden der Snappy-Bibliothek mit Amazon EMR

Snappy ist eine Komprimierungs- und Dekomprimierungsbibliothek, die für höhere Geschwindigkeit optimiert ist. Es ist in Amazon EMR AMIs Version 2.0 und höher verfügbar und wird als Standard für die Zwischenkomprimierung verwendet. Weitere Informationen zu Snappy finden Sie unter <http://code.google.com/p/snappy/>.

Planung und Konfiguration primärer Knoten in Ihrem Amazon EMR-Cluster

Wenn Sie einen Amazon-EMR-Cluster starten, können Sie festlegen, ob Ihr Cluster einen oder drei Primärknoten besitzt. Hochverfügbarkeit für Instance-Flotten wird mit den Amazon EMR-Versionen 5.36.1, 5.36.2, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12.0 und höher unterstützt. Hochverfügbarkeit für Instance-Gruppen wird mit Amazon-EMR-Versionen 5.23.0 und höher unterstützt. Um die Cluster-Verfügbarkeit weiter zu verbessern, kann Amazon EMR Amazon EC2 Placement-Gruppen verwenden, um sicherzustellen, dass die primären Knoten auf unterschiedlicher zugrunde liegender Hardware platziert werden. Weitere Informationen finden Sie unter [Amazon EMR-Integration mit EC2 Platzierungsgruppen](#).

Ein Amazon-EMR-Cluster mit mehreren Primärknoten bietet die folgenden Hauptvorteile:

- Der Primärknoten ist nicht länger eine einzelne Fehlerquelle. Wenn ein Primärknoten ausfällt, verwendet der Cluster die beiden anderen Primärknoten und wird ohne Unterbrechung weiter ausgeführt. In der Zwischenzeit ersetzt Amazon EMR den ausgefallenen Primärknoten automatisch durch einen neuen Primärknoten mit derselben Konfiguration und denselben Bootstrap-Aktionen.
- Amazon EMR aktiviert die Hadoop-Hochverfügbarkeitsfunktionen von HDFS NameNode und YARN ResourceManager und unterstützt Hochverfügbarkeit für einige andere Open-Source-Anwendungen.

Weitere Informationen darüber, wie ein Amazon-EMR-Cluster mit mehreren Primärknoten Open-Source-Anwendungen und andere Amazon-EMR-Features unterstützt, finden Sie unter [Funktionen, die Hochverfügbarkeit in einem Amazon EMR-Cluster unterstützen, und wie sie mit Open-Source-Anwendungen funktionieren](#).

 Note

Der Cluster kann sich nur in einer einzigen Availability Zone oder einem einzigen Subnetz befinden.

Dieser Abschnitt enthält Informationen zu unterstützten Anwendungen und Features eines Amazon-EMR-Clusters mit mehreren Primärknoten sowie Konfigurationsdetails, bewährte Methoden und Überlegungen zum Starten des Clusters.

Themen

- [Funktionen, die Hochverfügbarkeit in einem Amazon EMR-Cluster unterstützen, und wie sie mit Open-Source-Anwendungen funktionieren](#)
- [Amazon-EMR-Clustern mit mehreren Primärknoten starten](#)
- [Amazon EMR-Integration mit EC2 Platzierungsgruppen](#)
- [Überlegungen und bewährte Methoden bei der Erstellung eines Amazon EMR-Clusters mit mehreren Primärknoten](#)

Funktionen, die Hochverfügbarkeit in einem Amazon EMR-Cluster unterstützen, und wie sie mit Open-Source-Anwendungen funktionieren

Dieses Thema enthält Informationen zu den Hadoop-Hochverfügbarkeitsfunktionen von HDFS NameNode und YARN ResourceManager in einem Amazon EMR-Cluster und darüber, wie die Hochverfügbarkeitsfunktionen mit Open-Source-Anwendungen und anderen Amazon EMR-Funktionen funktionieren.

Hohe Verfügbarkeit – HDFS

Ein Amazon-EMR-Cluster mit mehreren Primärknoten aktiviert das Hochverfügbarkeitsfeature HDFS NameNode in Hadoop. Weitere Informationen finden Sie unter [HDFS – hohe Verfügbarkeit](#).

In einem Amazon EMR-Cluster sind zwei oder mehr separate Knoten konfiguriert als NameNodes. Einer NameNode befindet sich in einem `active` Staat und die anderen befinden sich in einem `standby` Staat. Wenn der Knoten mit `active` NameNode ausfällt, startet Amazon EMR einen automatischen HDFS-Failover-Prozess. Ein Knoten mit `standby` NameNode wird `active` und übernimmt alle Client-Operationen im Cluster. Amazon EMR ersetzt den Knoten durch einen neuen, der dann als `standby` erneut beitrifft.

 Note

In den Amazon EMR-Versionen 5.23.0 bis einschließlich 5.30.1 wird HDFS nur auf zwei der drei Primärknoten ausgeführt. NameNode

Wenn Sie herausfinden möchten, welcher NameNode `istactive`, können Sie SSH verwenden, um eine Verbindung zu einem beliebigen primären Knoten im Cluster herzustellen und den folgenden Befehl ausführen:

```
hdfs haadmin -getAllServiceState
```

In der Ausgabe werden die Knoten, auf denen installiert NameNode ist, und ihr Status aufgeführt. Zum Beispiel

```
ip-##-##-##1.ec2.internal:8020 active
ip-##-##-##2.ec2.internal:8020 standby
ip-##-##-##3.ec2.internal:8020 standby
```

Garn mit hoher Verfügbarkeit ResourceManager

Ein Amazon EMR-Cluster mit mehreren Primärknoten aktiviert die ResourceManager YARN-Hochverfügbarkeitsfunktion in Hadoop. Weitere Informationen finden Sie unter [ResourceManager Hochverfügbarkeit](#).

In einem Amazon EMR-Cluster mit mehreren Primärknoten ResourceManager wird YARN auf allen drei Primärknoten ausgeführt. Einer ResourceManager befindet sich im `active` Bundesstaat, und die anderen beiden befinden sich im `standby` Bundesstaat. Wenn der primäre Knoten mit `active` ResourceManager ausfällt, startet Amazon EMR einen automatischen Failover-Prozess. Ein primärer Knoten mit `a standby` ResourceManager übernimmt alle Operationen. Amazon EMR ersetzt den ausgefallenen Primärknoten durch einen neuen, der dann wieder dem ResourceManager Quorum als hinzugefügt wird. `standby`

Sie können für jeden primären Knoten eine Verbindung zu „[http: //:8088/clustermaster-public-dns-name](http://:8088/clustermaster-public-dns-name)“ herstellen, wodurch Sie automatisch zum Ressourcenmanager weitergeleitet werden. Um zu ermitteln, welcher ResourceManager den Zustand `active` hat, verwenden Sie SSH, um eine Verbindung mit einem Primärknoten im Cluster herzustellen. Führen Sie anschließend den folgenden Befehl aus, um die Liste der drei Primärknoten und deren Status abzurufen:

```
yarn rmadmin -getAllServiceState
```

Unterstützte Anwendungen in einem Amazon-EMR-Cluster mit mehreren Primärknoten

Sie können die folgenden Anwendungen auf einem Amazon-EMR-Cluster mit mehreren Primärknoten installieren und ausführen. Für jede Anwendung variiert der Failover-Prozess des Primärknotens.

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
Flink	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	Flink-Aufträge auf Amazon EMR werden als YARN-Anwendungen ausgeführt. Flink JobManagers läuft wie YARN auf den Kernknoten. ApplicationMasters Der JobManager wird durch den Failover-Prozess des primären Knotens nicht beeinflusst. Wenn Sie Amazon EMR Version 5.27.0 oder früher verwenden, JobManager handelt es sich um einen einzigen Fehlerpunkt. Wenn der JobManager fehlschlägt, gehen alle Job-Status verloren und die laufenden Jobs werden nicht wieder aufgenommen. Sie können JobManager Hochverfügbarkeit aktivieren, indem Sie die Anzahl der Anwendungsversuche, das Checkpointing und die Aktivierung ZooKeeper als Statusspeicher für Flink konfigurieren. Weitere Informationen finden Sie unter Konfigurieren von Flink auf einem Amazon-EMR-Cluster mit mehreren Primärknoten.

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
		Ab Amazon EMR Version 5.28.0 ist keine manuelle Konfiguration erforderlich, um Hochverfügbarkeit zu ermöglichen JobManager.
Ganglia	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	Ganglia ist auf allen Primärknoten verfügbar. Daher kann Ganglia während des Failover-Prozesses für den Primärknoten weiter ausgeführt werden.
Hadoop	Hohe Verfügbarkeit	HDFS NameNode und YARN schalten ResourceManager automatisch auf den Standby-Knoten um, wenn der aktive Primärknoten ausfällt.
HBase	Hohe Verfügbarkeit	HBase führt automatisch ein Failover zum Standby-Knoten durch, wenn der aktive Primärknoten ausfällt. Wenn Sie HBase über einen REST- oder Thrift-Server eine Verbindung herstellen, müssen Sie zu einem anderen Primärknoten wechseln, wenn der aktive Primärknoten ausfällt.
HCatalog	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	HCatalog basiert auf dem Hive-Metastore, der sich außerhalb des Clusters befindet. HCatalog bleibt während des Failover-Prozesses für den primären Knoten verfügbar.

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
JupyterHub	Hohe Verfügbarkeit	JupyterHub ist auf allen drei primären Instanzen installiert. Es wird dringend empfohlen, die Notebook-Persistenz zu konfigurieren, um Notebookverlust bei einem Ausfall des Primärknotens zu verhindern. Weitere Informationen finden Sie unter Konfigurieren von Persistenz für Notebooks in Amazon S3 .
Livy	Hohe Verfügbarkeit	Livy wird auf allen drei Primärknoten installiert. Bei einem Ausfall des aktiven Primärknotens verlieren Sie den Zugriff auf die aktuelle Livy-Sitzung und müssen eine neue Livy-Sitzung auf einem anderen Primärknoten oder auf dem neuen Ersatzknoten erstellen.
Mahout	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	Da Mahout keinen Daemon besitzt, hat der Failover-Prozess für den Primärknoten keine Auswirkungen.
MXNet	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	Da MXNet es keinen Daemon gibt, ist er vom Failover-Prozess des Primärknotens nicht betroffen.
Phoenix	Hochverfügbarkeit	Phoenix QueryServer läuft nur auf einem der drei Primärknoten. Phoenix ist auf allen drei Mastern so konfiguriert, dass er den Phoenix verbindet. QueryServer Sie können die private IP des Phoenix QueryServer anhand der <code>/etc/phoenix/conf/phoenix-env.sh</code> - Datei finden

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
Pig	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	Da Pig keinen Daemon besitzt, hat der Failover-Prozess für den Primärknoten keine Auswirkungen.
Spark	Hohe Verfügbarkeit	Alle Spark-Anwendungen werden in YARN-Containern ausgeführt und reagieren auf den Failover-Prozess für einen Primärknoten auf die gleiche Weise wie YARN-Hochverfügbarkeitsfeatures.
Sqoop	Hohe Verfügbarkeit	Standardmäßig speichern sqoop-job und sqoop-metastore Daten (Auftragsbeschreibungen) auf der lokalen Festplatte des Masters, der den Befehl ausführt. Wenn Sie Metastore-Daten in einer externen Datenbank speichern möchten, schlagen Sie bitte in der Apache Sqoop-Dokumentation nach.
Tez	Hohe Verfügbarkeit	Da Tez Container auf YARN ausgeführt werden, verhält sich Tez während des Failover-Prozesses für den Primärknoten wie YARN.
TensorFlow	Verfügbarkeit nicht durch Failover für den Primärknoten betroffen	Da TensorFlow es keinen Daemon gibt, ist er vom Failover-Prozess des Primärknotens nicht betroffen.

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
Zeppelin	Hohe Verfügbarkeit	Zeppelin wird auf allen drei Primärknoten installiert. Zeppelin speichert Notizen und Interpreterkonfigurationen standardmäßig in HDFS, um Datenverlust zu verhindern. Interpreter ersetzungen sind über alle drei Primär-Instances vollständig isoliert. Sitzungsdaten gehen beim Master-Ausfall verloren. Es wird empfohlen, dieselbe Notiz nicht gleichzeitig auf verschiedenen Primär-Instances zu ändern.
ZooKeeper	Hohe Verfügbarkeit	ZooKeeper ist die Grundlage der automatischen HDFS-Failover-Funktion. ZooKeeper bietet einen hochverfügbaren Dienst zur Verwaltung von Koordinationsdaten, zur Benachrichtigung von Clients über Änderungen an diesen Daten und zur Überwachung von Clients im Hinblick auf Ausfälle. Weitere Informationen finden Sie unter HDFS-Funktion für automatische Failover .

Um die folgenden Anwendungen in einem Amazon-EMR-Cluster mit mehreren Primärknoten auszuführen, müssen Sie eine externe Datenbank konfigurieren. Die externe Datenbank befindet sich außerhalb des Clusters. Daher sind Daten während des Failover-Prozesses für den Primärknoten persistent. Für die folgenden Anwendungen werden die Servicekomponenten während des Primärknoten-Failoverprozesses automatisch wiederhergestellt, aktive Aufträge können jedoch fehlschlagen und müssen erneut versucht werden.

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
Hive	Hohe Verfügbarkeit, jedoch ausschließlich für Service-Komponenten	Ein externer Metastore für Hive ist erforderlich. Dies muss ein externer MySQL-Metastore

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
		sein, da PostgreSQL für Multi-Master-Cluster nicht unterstützt wird. Weiter Informationen finden Sie unter Konfigurieren eines externen Metastores für Hive .
Hue	Hohe Verfügbarkeit, jedoch ausschließlich für Service-Komponenten	Eine externe Datenbank für Hue ist erforderlich. Weitere Informationen finden Sie unter Verwenden von Hue mit einer Remote-Datenbank in Amazon RDS .
Oozie	Hohe Verfügbarkeit, jedoch ausschließlich für Service-Komponenten	Eine externe Datenbank für Oozie ist erforderlich. Weitere Informationen finden Sie unter Verwenden von Oozie mit einer Remote-Datenbank in Amazon RDS. Oozie-Server und Oozie-Client sind auf allen drei Primärknoten installiert. Die oozie-clients sind so konfiguriert, dass sie standardmäßig eine Verbindung mit dem richtigen oozie-server herstellen.

Anwendung	Verfügbarkeit während Failover für den Primärknoten	Hinweise
PrestoDB oder PrestoSQL/Trino	Hohe Verfügbarkeit, jedoch ausschließlich für Service-Komponenten	Ein externer Hive-Metastore für PrestoDB (PrestoSQL auf Amazon EMR 6.1.0-6.3.0 oder Trino auf Amazon EMR 6.4.0 und höher) ist erforderlich. Sie können Presto mit dem AWS Glue Data Catalog oder eine externe MySQL-Datenbank für Hive verwenden. Die Presto-CLI ist auf allen drei Primärknoten installiert, sodass Sie damit von jedem der Primärknoten aus auf den Presto Coordinator zugreifen können. Der Presto Coordinator ist nur auf einem Primärknoten installiert. Sie können den DNS-Namen des Primärknotens ermitteln, auf dem der Presto Coordinator installiert ist, indem Sie die <code>Amazon-describe-cluster -EMR-API</code> aufrufen und den zurückgegebenen Wert des <code>MasterPublicDnsName</code> -Felds in der Antwort lesen.

Note

Beim Ausfall eines Primärknoten wird die Verbindung Ihrer Java Database Connectivity (JDBC) oder Open Database Connectivity (ODBC) mit dem Primärknoten beendet. Sie können eine Verbindung mit einem der verbleibenden Primärknoten herstellen und Ihre Arbeit fortsetzen, da der Hive-Metastore-Daemon auf allen Primärknoten ausgeführt wird. Sie können auch warten, bis der ausgefallene Primärknoten ersetzt wird.

So funktionieren Amazon-EMR-Feature in einem Cluster mit mehreren Primärknoten

Herstellen von Verbindungen mit Primärknoten über SSH

Sie können mit jedem der drei Primärknoten in einem Amazon-EMR-Cluster auf die gleiche Art Verbindungen über SSH herstellen, wie Sie Verbindungen mit einem einzelnen Primärknoten herstellen. Weitere Informationen finden Sie unter [Mit dem Primärknoten über SSH verbinden](#).

Beim Ausfall eines Primärknotens wird Ihre Verbindung mit diesem Primärknoten beendet. Um Ihre Arbeit fortzusetzen, können Sie eine Verbindung mit einem der beiden anderen Primärknoten herstellen. Alternativ können Sie auf den neuen Primärknoten zugreifen, nachdem Amazon EMR den ausgefallenen Primärknoten durch einen neuen Primärknoten ersetzt hat.

Note

Die private IP-Adresse des ersetzenden Primärknoten ist mit der privaten IP-Adresse des vorherigen Primärknotens identisch. Die öffentliche IP-Adresse des ersetzenden Primärknotens wird möglicherweise geändert. Sie können die neue IP-Adressen in der Konsole oder über den Befehl `describe-cluster` in der AWS -CLI abrufen. NameNode läuft nur auf zwei der primären Knoten. Sie können jedoch `hdfs`-CLI-Befehle ausführen und Aufgaben ausführen, um auf allen drei Primärknoten auf HDFS zuzugreifen.

Arbeiten mit Schritten in einem Amazon-EMR-Cluster mit mehreren Primärknoten

Sie können Schritte an einen Amazon-EMR-Cluster mit mehreren Primärknoten auf die gleiche Weise übermitteln, wie Sie mit Schritten in einem Cluster mit einem einzelnen Primärknoten arbeiten. Weitere Informationen finden Sie unter [Übermitteln von Arbeit an einen Cluster](#).

Im Folgenden finden Sie Überlegungen zur Arbeit mit Schritten in einem Amazon-EMR-Cluster mit mehreren Primärknoten:

- Beim Ausfall eines Primärknotens werden die Schritte, die auf dem Primärknoten ausgeführt werden, als FAILED (fehlgeschlagen) markiert. Alle lokal geschriebenen Daten gehen verloren. Der Status FAILED gibt jedoch möglicherweise nicht den tatsächlichen Zustand der Schritte wider.
- Wenn ein Schritt, der ausgeführt wurde, als der Primärknoten ausfiel, eine YARN-Anwendung gestartet hatte, kann der Schritt aufgrund des automatischen Failover-Prozesses für den Primärknoten weiter ausgeführt und erfolgreich abgeschlossen werden.

- Sie sollten den Status von Schritten anhand der Ausgaben der Aufgaben überprüfen. MapReduce Jobs verwenden beispielsweise eine `_SUCCESS` Datei, um festzustellen, ob der Job erfolgreich abgeschlossen wurde.
- Es wird empfohlen, den `ActionOnFailure` Parameter auf `CONTINUE` oder `CANCEL_AND_WAIT` statt auf `TERMINATE_JOB_FLOW` oder `TERMINATE_CLUSTER` festzulegen.

Automatischer Beendigungsschutz

Amazon EMR aktiviert automatisch den Kündigungsschutz für alle Cluster mit mehreren Primärknoten und überschreibt alle Einstellungen für die Schrittausführung, die Sie bei der Erstellung des Clusters angeben. Sie können den Kündigungsschutz deaktivieren, nachdem der Cluster gestartet wurde. Siehe [Konfigurieren des Beendigungsschutzes für aktive Cluster](#). Um einen Cluster mit mehreren Primärknoten herunterzufahren, müssen Sie zunächst die Clusterattribute ändern, um den Kündigungsschutz zu deaktivieren. Detaillierte Anweisungen finden Sie unter [Amazon-EMR-Clustern mit mehreren Primärknoten beenden](#).

Weitere Informationen zum Beendigungsschutz finden Sie unter [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).

Nicht unterstützte Feature in einem Amazon-EMR-Cluster mit mehreren Primärknoten

Die folgenden Amazon-EMR-Feature sind derzeit in einem Amazon-EMR-Cluster mit mehreren Primärknoten nicht verfügbar:

- EMR Notebooks
- Zugriff auf den permanenten Spark History Server mit nur einem Klick
- Persistente Anwendungsbenutzeroberflächen
- Der Ein-Klick-Zugriff auf persistente Anwendungsbenutzeroberflächen ist derzeit nicht für Amazon EMR-Cluster mit mehreren Primärknoten oder für in Lake Formation integrierte Amazon EMR-Cluster verfügbar. AWS

Note

Um in Ihrem Cluster Kerberos-Authentifizierung zu verwenden, müssen Sie einen externen KDC konfigurieren.

Ab Amazon-EMR-Version 5.27.0 können Sie die transparente HDFS-Verschlüsselung auf einem Amazon-EMR-Cluster mit mehreren Primärknoten konfigurieren. Weitere Informationen finden Sie unter [Transparente Verschlüsselung in HDFS in Amazon EMR](#).

Amazon-EMR-Clustern mit mehreren Primärknoten starten

Dieses Thema enthält Konfigurationsdetails und Beispiele für den Start eines Amazon-EMR-Clusters mit mehreren Primärknoten.

Note

Amazon EMR aktiviert automatisch den Kündigungsschutz für alle Cluster mit mehreren Primärknoten und überschreibt alle Einstellungen für die automatische Terminierung, die Sie bei der Erstellung des Clusters angeben. Um einen Cluster mit mehreren Primärknoten herunterzufahren, müssen Sie zunächst die Clusterattribute ändern, um den Kündigungsschutz zu deaktivieren. Detaillierte Anweisungen finden Sie unter [Amazon-EMR-Clustern mit mehreren Primärknoten beenden](#).

Voraussetzungen

- Sie können einen Amazon-EMR-Cluster mit mehreren Primärknoten in öffentlichen und privaten VPC-Subnetzen starten. EC2-Classic wird nicht unterstützt. Um einen Amazon EMR-Cluster mit mehreren Primärknoten in einem öffentlichen Subnetz zu starten, müssen Sie den Instances in diesem Subnetz den Empfang einer öffentlichen IP-Adresse ermöglichen, indem Sie IPv4 in der Konsole Automatisch zuweisen auswählen oder den folgenden Befehl ausführen. Ersetzen Sie es durch Ihre **22XXXX01** Subnetz-ID.

```
aws ec2 modify-subnet-attribute --subnet-id subnet-22XXXX01 --map-public-ip-on-launch
```

- Um Hive, Hue oder Oozie auf einem Amazon-EMR-Cluster mit mehreren Primärknoten auszuführen, müssen Sie einen externen Metastore erstellen. Weitere Informationen finden Sie unter [Konfigurieren eines externen Metastores für Hive](#), [Verwenden von Hue mit einer Remote-Datenbank in Amazon RDS](#) oder [Apache Oozie](#).

- Um in Ihrem Cluster Kerberos-Authentifizierung zu verwenden, müssen Sie einen externen KDC konfigurieren. Weitere Informationen finden Sie unter [Konfigurieren von Kerberos auf Amazon EMR](#).

Amazon-EMR-Clustern mit mehreren Primärknoten starten

Sie können einen Cluster mit mehreren Primärknoten starten, wenn Sie Instance-Gruppen oder Instance-Flotten verwenden. Wenn Sie Instance-Gruppen mit mehreren Primärknoten verwenden, müssen Sie für die Primärknoten-Instance-Gruppe den Wert 3 für die Zahl der Instances angeben. Wenn Sie Instance-Flotten mit mehreren Primärknoten verwenden, müssen Sie die TargetOnDemandCapacity von 3, die TargetSpotCapacity von 0 für die primäre Instance-Flotte und die WeightedCapacity von 1 für jeden Instance-Typ angeben, den Sie für die primäre Flotte konfigurieren.

Die folgenden Beispiele zeigen, wie Sie den Cluster mit einem Standard- oder benutzerdefinierten AMI mit sowohl Instance-Gruppen als auch Instance-Flotten starten:

Note

Sie müssen die Subnetz-ID angeben, wenn Sie einen Amazon-EMR-Cluster mit mehreren Primärknoten mithilfe der AWS CLI starten. Ersetzen Sie in **22XXXX02** den folgenden Beispielen **22XXXX01** und durch Ihre Subnetz-ID.

Default AMI, instance groups

Example Beispiel – Starten eines Instance-Gruppen-Clusters von Amazon EMR mit mehreren Primärknoten unter Verwendung eines Standard-AMI

```
aws emr create-cluster \  
--name "ha-cluster" \  
--release-label emr-6.15.0 \  
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge \  
InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \  
--ec2-attributes \  
KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01 \  
\  
--service-role EMR_DefaultRole \  
--applications Name=Hadoop Name=Spark
```

Default AMI, instance fleets

Example Beispiel – Starten eines Instance-Flotten-Clusters von Amazon EMR mit mehreren Primärknoten unter Verwendung eines Standard-AMI

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        }
    }
]
```

```

    }
  },
  "InstanceTypeConfigs": [
    {
      "WeightedCapacity": 1,
      "BidPriceAsPercentageOfOnDemandPrice": 100,
      "InstanceType": "m5.xlarge"
    },
    {
      "WeightedCapacity": 2,
      "BidPriceAsPercentageOfOnDemandPrice": 100,
      "InstanceType": "m5.2xlarge"
    },
    {
      "WeightedCapacity": 4,
      "BidPriceAsPercentageOfOnDemandPrice": 100,
      "InstanceType": "m5.4xlarge"
    }
  ],
  "Name": "Core - 2"
}
]' \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":
["subnet-22XXXX01", "subnet-22XXXX02"]}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark

```

Custom AMI, instance groups

Example Beispiel – Starten eines Instance-Gruppen-Clusters von Amazon EMR mit mehreren Primärknoten unter Verwendung eines benutzerdefinierten AMI

```

aws emr create-cluster \
--name "custom-ami-ha-cluster" \
--release-label emr-6.15.0 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
\
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID

```

Custom AMI, instance fleets

Example Beispiel – Starten eines Instance-Flotten-Clusters von Amazon EMR mit mehreren Primärknoten unter Verwendung eines benutzerdefinierten AMI

```
aws emr create-cluster \  
--name "ha-cluster" \  
--release-label emr-6.15.0 \  
--instance-fleets '[  
  {  
    "InstanceFleetType": "MASTER",  
    "TargetOnDemandCapacity": 3,  
    "TargetSpotCapacity": 0,  
    "LaunchSpecifications": {  
      "OnDemandSpecification": {  
        "AllocationStrategy": "lowest-price"  
      }  
    },  
    "InstanceTypeConfigs": [  
      {  
        "WeightedCapacity": 1,  
        "BidPriceAsPercentageOfOnDemandPrice": 100,  
        "InstanceType": "m5.xlarge"  
      },  
      {  
        "WeightedCapacity": 1,  
        "BidPriceAsPercentageOfOnDemandPrice": 100,  
        "InstanceType": "m5.2xlarge"  
      },  
      {  
        "WeightedCapacity": 1,  
        "BidPriceAsPercentageOfOnDemandPrice": 100,  
        "InstanceType": "m5.4xlarge"  
      }  
    ],  
    "Name": "Master - 1"  
  },  
  {  
    "InstanceFleetType": "CORE",  
    "TargetOnDemandCapacity": 5,  
    "TargetSpotCapacity": 0,  
    "LaunchSpecifications": {  
      "OnDemandSpecification": {  
        "AllocationStrategy": "lowest-price"
```

```

    }
  },
  "InstanceTypeConfigs": [
    {
      "WeightedCapacity": 1,
      "BidPriceAsPercentageOfOnDemandPrice": 100,
      "InstanceType": "m5.xlarge"
    },
    {
      "WeightedCapacity": 2,
      "BidPriceAsPercentageOfOnDemandPrice": 100,
      "InstanceType": "m5.2xlarge"
    },
    {
      "WeightedCapacity": 4,
      "BidPriceAsPercentageOfOnDemandPrice": 100,
      "InstanceType": "m5.4xlarge"
    }
  ],
  "Name": "Core - 2"
}
]' \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":
["subnet-22XXXX01", "subnet-22XXXX02"]}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID

```

Amazon-EMR-Clustern mit mehreren Primärknoten beenden

Um einen Amazon-EMR-Cluster mit mehreren Primärknoten zu beenden, müssen Sie den Beendigungsschutz deaktivieren, bevor Sie den Cluster beenden, wie das folgende Beispiel zeigt. Ersetzen Sie **j-3KVTXXXXXX7UG** durch die ID Ihres Clusters.

```

aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
aws emr terminate-clusters --cluster-id j-3KVTXXXXXX7UG

```

Amazon EMR-Integration mit EC2 Platzierungsgruppen

Wenn Sie einen Amazon EMR-Cluster mit mehreren primären Knoten auf Amazon starten EC2, haben Sie die Möglichkeit, Platzierungsgruppenstrategien zu verwenden, um festzulegen, wie die Primärknoten-Instances zum Schutz vor Hardwareausfällen bereitgestellt werden sollen.

Platzierungsgruppenstrategien werden ab Amazon-EMR-Version 5.23.0 als Option für Cluster mit mehreren Primärknoten unterstützt. Derzeit werden nur Primärknotentypen von der Platzierungsgruppenstrategie unterstützt, und die SPREAD-Strategie wird auf diese Primärknoten angewendet. Bei dieser SPREAD-Strategie wird eine kleine Gruppe von Instances auf separater zugrundeliegender Hardware platziert, um den Verlust mehrerer Primärknoten im Falle eines Hardwarefehlers zu verhindern. Beachten Sie, dass eine Anforderung zum Starten einer Instance fehlschlagen kann, wenn es nicht genügend eindeutige Hardware zur Erfüllung der Anforderung gibt. Weitere Informationen zu EC2 Platzierungsstrategien und Einschränkungen finden Sie unter [Platzierungsgruppen](#) im EC2 Benutzerhandbuch für Linux-Instances.

Amazon gibt ein anfängliches Limit EC2 von 500 Clustern mit aktivierter Platzierungsgruppenstrategie, die pro AWS Region gestartet werden können. Wenden Sie sich an den AWS Support, um eine Erhöhung der Anzahl der zulässigen Platzierungsgruppen zu beantragen. Sie können EC2 Platzierungsgruppen identifizieren, die Amazon EMR erstellt, indem Sie das Schlüssel-Wert-Paar verfolgen, das Amazon EMR mit der Amazon EMR-Platzierungsgruppenstrategie verknüpft. Weitere Informationen zu EC2 Cluster-Instance-Tags finden Sie unter: [Cluster-Instances in Amazon anzeigen EC2](#)

Hängen Sie die von der Platzierungsgruppe verwaltete Richtlinie an Amazon an EMRrole

Die Platzierungsgruppenstrategie erfordert eine verwaltete Richtlinie namens `AmazonElasticMapReducePlacementGroupPolicy`, die es Amazon EMR ermöglicht, Platzierungsgruppen auf Amazon EC2 zu erstellen, zu löschen und zu beschreiben. Sie müssen `AmazonElasticMapReducePlacementGroupPolicy` an die Servicerolle für Amazon EMR anhängen, bevor Sie einen Amazon-EMR-Cluster mit mehreren Primärknoten starten.

Sie können die `AmazonEMRServicePolicy_v2` verwaltete Richtlinie alternativ der Amazon EMR-Servicerolle anstelle der verwalteten Richtlinie für die Platzierungsgruppe zuordnen. `AmazonEMRServicePolicy_v2` ermöglicht den gleichen Zugriff auf Platzierungsgruppen bei Amazon EC2 wie der `AmazonElasticMapReducePlacementGroupPolicy`. Weitere Informationen finden Sie unter [Servicerolle für Amazon EMR \(EMR-Rolle\)](#).

Bei der von AmazonElasticMapReducePlacementGroupPolicy verwalteten Richtlinie handelt es sich um den folgenden JSON-Text, der von Amazon EMR erstellt und verwaltet wird.

 Note

Da die AmazonElasticMapReducePlacementGroupPolicy verwaltete Richtlinie automatisch aktualisiert wird, kann es sein, dass die hier gezeigte Richtlinie aktualisiert wird out-of-date. Verwenden Sie die AWS Management Console, um die aktuelle Richtlinie einzusehen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "ec2:DeletePlacementGroup",
        "ec2:DescribePlacementGroups"
      ]
    },
    {
      "Resource": "arn:aws:ec2:*:*:placement-group/pg-*",
      "Effect": "Allow",
      "Action": [
        "ec2:CreatePlacementGroup"
      ]
    }
  ]
}
```

Einen Amazon-EMR-Cluster mit mehrere Primärknoten mittels Platzierungsgruppenstrategie starten

Um einen Amazon-EMR-Cluster, der mehrere Primärknoten hat, mit einer Platzierungsgruppenstrategie zu starten, fügen Sie der Amazon-EMR-Rolle die von der Platzierungsgruppe verwaltete Richtlinie AmazonElasticMapReducePlacementGroupPolicy hinzu. Weitere Informationen finden Sie unter [Hängen Sie die von der Platzierungsgruppe verwaltete Richtlinie an Amazon an EMRrole](#).

Jedes Mal, wenn Sie diese Rolle verwenden, um einen Amazon-EMR-Cluster mit mehreren Primärknoten zu starten, versucht Amazon EMR, einen Cluster zu starten, wobei die SPREAD-Strategie auf seine Primärknoten angewendet wird. Wenn Sie eine Rolle verwenden, der nicht die von der Platzierungsgruppe verwaltete Richtlinie `AmazonElasticMapReducePlacementGroupPolicy` zugeordnet ist, versucht Amazon EMR, einen Amazon-EMR-Cluster mit mehreren Primärknoten ohne Platzierungsgruppenstrategie zu starten.

Wenn Sie einen Amazon EMR-Cluster mit mehreren Primärknoten starten, wobei der `placement-group-configs` Parameter Amazon EMR API oder CLI verwendet, startet Amazon EMR den Cluster nur, wenn Amazon die Richtlinie für die Verwaltung der Platzierungsgruppe zugewiesen `EMRrole` hat. `AmazonElasticMapReducePlacementGroupPolicy` Wenn Amazon die Richtlinie `EMRrole` nicht angehängt hat, schlägt der Start des Amazon EMR-Clusters mit mehreren Primärknoten fehl.

Amazon EMR API

Example Beispiel – Verwenden Sie eine Platzierungsgruppenstrategie, um einen Instance-Gruppen-Cluster mit mehreren Primärknoten über die Amazon-EMR-API zu starten

Wenn Sie die `RunJobFlow` Aktion verwenden, um einen Amazon EMR-Cluster mit mehreren Primärknoten zu erstellen, legen Sie die `PlacementGroupConfigs` Eigenschaft wie folgt fest. Derzeit wird die MASTER-Instance-Rolle automatisch SPREAD als Platzierungsgruppenstrategie verwendet.

```
{
  "Name": "ha-cluster",
  "PlacementGroupConfigs": [
    {
      "InstanceRole": "MASTER"
    }
  ],
  "ReleaseLabel": "emr-6.15.0",
  "Instances": {
    "ec2SubnetId": "subnet-22XXXX01",
    "ec2KeyName": "ec2_key_pair_name",
    "InstanceGroups": [
      {
        "InstanceCount": 3,
        "InstanceRole": "MASTER",
        "InstanceType": "m5.xlarge"
      }
    ],
  },
}
```



```

        {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
        }
    ],
    "JobFlowRole":"EMR_EC2_DefaultRole",
    "ServiceRole":"EMR_DefaultRole"
}

```

- Ersetzen Sie es *ha-cluster* durch den Namen Ihres Hochverfügbarkeitsclusters.
- Ersetzen Sie *subnet-22XXX01* durch Ihre Subnetz-ID.
- Ersetzen Sie das *ec2_key_pair_name* durch den Namen Ihres EC2 key pair für diesen Cluster. EC2 Ein key pair ist optional und nur erforderlich, wenn Sie SSH für den Zugriff auf Ihren Cluster verwenden möchten.

AWS CLI

Example Beispiel – Verwenden Sie eine Platzierungsgruppenstrategie, um einen Instance-Flotten-Cluster mit mehreren Primärknoten über die AWS Command Line Interface zu starten

Wenn Sie die RunJobFlow Aktion verwenden, um einen Amazon EMR-Cluster mit mehreren Primärknoten zu erstellen, legen Sie die PlacementGroupConfigs Eigenschaft wie folgt fest. Derzeit wird die MASTER-Instance-Rolle automatisch SPREAD als Platzierungsgruppenstrategie verwendet.

```

aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        }
    },

```

```

    "InstanceTypeConfigs": [
      {
        "WeightedCapacity": 1,
        "BidPriceAsPercentageOfOnDemandPrice": 100,
        "InstanceType": "m5.xlarge"
      },
      {
        "WeightedCapacity": 1,
        "BidPriceAsPercentageOfOnDemandPrice": 100,
        "InstanceType": "m5.2xlarge"
      },
      {
        "WeightedCapacity": 1,
        "BidPriceAsPercentageOfOnDemandPrice": 100,
        "InstanceType": "m5.4xlarge"
      }
    ],
    "Name": "Master - 1"
  },
  {
    "InstanceFleetType": "CORE",
    "TargetOnDemandCapacity": 5,
    "TargetSpotCapacity": 0,
    "LaunchSpecifications": {
      "OnDemandSpecification": {
        "AllocationStrategy": "lowest-price"
      }
    },
    "InstanceTypeConfigs": [
      {
        "WeightedCapacity": 1,
        "BidPriceAsPercentageOfOnDemandPrice": 100,
        "InstanceType": "m5.xlarge"
      },
      {
        "WeightedCapacity": 2,
        "BidPriceAsPercentageOfOnDemandPrice": 100,
        "InstanceType": "m5.2xlarge"
      },
      {
        "WeightedCapacity": 4,
        "BidPriceAsPercentageOfOnDemandPrice": 100,
        "InstanceType": "m5.4xlarge"
      }
    ]
  }

```

```

    ],
    "Name": "Core - 2"
  }
]' \
--ec2-attributes '{
  "KeyName": "ec2_key_pair_name",
  "InstanceProfile": "EMR_EC2_DefaultRole",
  "SubnetIds": [
    "subnet-22XXXX01",
    "subnet-22XXXX02"
  ]
}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark

```

- Ersetzen Sie es *ha-cluster* durch den Namen Ihres Hochverfügbarkeitsclusters.
- Ersetzen Sie das *ec2_key_pair_name* durch den Namen Ihres EC2 key pair für diesen Cluster. EC2 Ein key pair ist optional und nur erforderlich, wenn Sie SSH für den Zugriff auf Ihren Cluster verwenden möchten.
- Ersetzen Sie *subnet-22XXXX01* und *subnet-22XXXX02* durch Ihr IDs Subnetz.

Starten Sie einen Cluster mit mehreren Primärknoten ohne eine Platzierungsgruppenstrategie

Damit ein Cluster mit mehreren Primärknoten ohne die Platzierungsgruppenstrategie starten kann, müssen Sie einen der folgenden Schritte ausführen:

- Entfernen Sie die von der Platzierungsgruppe verwaltete Richtlinie `AmazonElasticMapReducePlacementGroupPolicy` aus `Amazon EMRole`, oder
- Starten Sie einen Cluster mit mehreren Primärknoten, wobei der `placement-group-configs`-Parameter `Amazon EMRAPI` oder `CLI NONE` als Platzierungsgruppenstrategie verwendet.

Amazon EMR API

Example – Starten eines Clusters mit mehreren Primärknoten ohne Platzierungsgruppenstrategie mithilfe der Amazon EMRAPI.

Wenn Sie die `RunJobFlow` Aktion verwenden, um einen Cluster mit mehreren Primärknoten zu erstellen, legen Sie die `PlacementGroupConfigs` Eigenschaft wie folgt fest.

```
{
  "Name": "ha-cluster",
  "PlacementGroupConfigs": [
    {
      "InstanceRole": "MASTER",
      "PlacementStrategy": "NONE"
    }
  ],
  "ReleaseLabel": "emr-5.30.1",
  "Instances": {
    "ec2SubnetId": "subnet-22XXXX01",
    "ec2KeyName": "ec2_key_pair_name",
    "InstanceGroups": [
      {
        "InstanceCount": 3,
        "InstanceRole": "MASTER",
        "InstanceType": "m5.xlarge"
      },
      {
        "InstanceCount": 4,
        "InstanceRole": "CORE",
        "InstanceType": "m5.xlarge"
      }
    ]
  },
  "JobFlowRole": "EMR_EC2_DefaultRole",
  "ServiceRole": "EMR_DefaultRole"
}
```

- *ha-cluster* Ersetzen Sie es durch den Namen Ihres Hochverfügbarkeitsclusters.
- Ersetzen Sie *subnet-22XXXX01* durch Ihre Subnetz-ID.
- Ersetzen Sie das *ec2_key_pair_name* durch den Namen Ihres EC2 key pair für diesen Cluster. EC2 Ein key pair ist optional und nur erforderlich, wenn Sie SSH für den Zugriff auf Ihren Cluster verwenden möchten.

Amazon EMR CLI

Example – Starten eines Clusters mit mehreren Primärknoten ohne Platzierungsgruppenstrategie mithilfe der Amazon EMR CLI.

Wenn Sie die RunJobFlow Aktion verwenden, um einen Cluster mit mehreren Primärknoten zu erstellen, legen Sie die PlacementGroupConfigs Eigenschaft wie folgt fest.

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER,PlacementStrategy=NONE \
--release-label emr-5.30.1 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
\
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

- *ha-cluster* Ersetzen Sie es durch den Namen Ihres Hochverfügbarkeitsclusters.
- Ersetzen Sie *subnet-22XXXX01* durch Ihre Subnetz-ID.
- Ersetzen Sie das *ec2_key_pair_name* durch den Namen Ihres EC2 key pair für diesen Cluster. EC2 Ein key pair ist optional und nur erforderlich, wenn Sie SSH für den Zugriff auf Ihren Cluster verwenden möchten.

Überprüfen Sie die Konfiguration der Platzierungsgruppenstrategie, die an den Cluster mit mehreren Primärknoten angehängt ist

Sie können die Amazon-EMR-Cluster-API verwenden, um die Konfiguration der Platzierungsgruppenstrategie anzuzeigen, die dem Cluster mit mehreren Primärknoten zugeordnet ist.

Example

```
aws emr describe-cluster --cluster-id "j-xxxxx"
{
  "Cluster":{
    "Id":"j-xxxxx",
    ...
  }
}
```

```
...
    "PlacementGroups":[
        {
            "InstanceRole":"MASTER",
            "PlacementStrategy":"SPREAD"
        }
    ]
}
```

Überlegungen und bewährte Methoden bei der Erstellung eines Amazon EMR-Clusters mit mehreren Primärknoten

Beachten Sie beim Erstellen eines Amazon-EMR-Clusters mit mehreren Primärknoten Folgendes:

Important

Um EMR-Cluster mit hoher Verfügbarkeit mit mehreren Primärknoten zu starten, empfehlen wir dringend, die neueste Amazon-EMR-Version zu verwenden. Dadurch wird sichergestellt, dass Sie ein Höchstmaß an Resilienz und Stabilität für Ihre Hochverfügbarkeits-Cluster erhalten.

- Hochverfügbarkeit für Instance-Flotten wird mit den Amazon EMR-Versionen 5.36.1, 5.36.2, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12.0 und höher unterstützt. Hochverfügbarkeit für Instance-Gruppen wird mit Amazon-EMR-Versionen 5.23.0 und höher unterstützt. Weitere Informationen finden Sie unter [Über Amazon-EMR-Releases](#).
- Auf Clustern mit hoher Verfügbarkeit unterstützt Amazon EMR nur den Start von Primärknoten mit On-Demand-Instances. Dadurch wird die höchste Verfügbarkeit für Ihren Cluster gewährleistet.
- Sie können immer noch mehrere Instance-Typen für die primäre Flotte angeben, aber alle Primärknoten von Hochverfügbarkeits-Clustern werden mit demselben Instance-Typ gestartet, einschließlich Ersatz-Instances für fehlerhafte Primärknoten.
- Um den Betrieb fortzusetzen, müssen bei einem Hochverfügbarkeits-Cluster mit mehreren Primärknoten zwei von drei Primärknoten fehlerfrei sein. Wenn also zwei Primärknoten gleichzeitig ausfallen, fällt Ihr EMR-Cluster aus.
- Alle EMR-Cluster, einschließlich Hochverfügbarkeits-Cluster, werden in einer einzigen Availability Zone gestartet. Daher können sie Ausfälle in der Availability Zone nicht tolerieren. Beim Ausfall einer Availability Zone verlieren Sie den Zugriff auf den Cluster.

- Wenn Sie beim Starten eines Clusters innerhalb einer Instance-Flotte eine benutzerdefinierte Servicerolle oder -richtlinie verwenden, können Sie die `ec2:DescribeInstanceTypeOfferings` Berechtigung hinzufügen, damit Amazon EMR nicht unterstützte Availability Zones (AZ) herausfiltern kann. Wenn Amazon EMR diejenigen herausfiltert AZs, die keine Instance-Typen von Primärknoten unterstützen, verhindert Amazon EMR, dass Cluster-Starts aufgrund nicht unterstützter primärer Instance-Typen fehlschlagen. Weitere Informationen finden Sie unter [Instance-Typ](#) wird nicht unterstützt.
- Amazon EMR garantiert keine Hochverfügbarkeit von Open-Source-Anwendungen, die nicht in [Unterstützte Anwendungen in einem Amazon-EMR-Cluster mit mehreren Primärknoten](#) angegeben sind.
- In den Amazon EMR-Versionen 5.23.0 bis 5.36.2 werden nur zwei der drei primären Knoten für einen Instance-Gruppen-Cluster ausgeführt. HDFS NameNode
- In Amazon EMR-Versionen 6.x und höher werden alle drei primären Knoten für eine Instance-Gruppe ausgeführt. HDFS NameNode

Überlegungen für das Konfigurieren von Subnetzen:

- Ein Amazon-EMR-Cluster mit mehreren Primärknoten kann sich nur in einer Availability Zone oder einem Subnetz befinden. Amazon EMR kann einen ausgefallenen Primärknoten nicht ersetzen, wenn das Subnetz zum Zeitpunkt des Failover-Prozesses vollständig ausgelastet oder überabonniert ist. Um dieses Szenario zu vermeiden, sollten Sie für einen Amazon EMR-Cluster ein vollständiges Subnetz reservieren. Darüber hinaus sollten Sie sicherstellen, dass im Subnetz eine ausreichende Zahl von privaten IP-Adressen verfügbar ist.

Überlegungen für das Konfigurieren von Core-Knoten:

- Um sicherzustellen, dass die Core-Knoten ebenfalls hoch verfügbar sind, sollten Sie mindestens vier Core-Knoten starten. Wenn Sie sich entscheiden, einen kleineren Cluster mit drei oder weniger Core-Knoten zu starten, legen Sie `dfs.replication` parameter auf mindestens 2 fest, damit HDFS über eine ausreichende DFS-Replikation verfügt. Weitere Informationen finden Sie unter [HDFS-Konfiguration](#).

 Warning

1. Das Festlegen von `dfs.replication` auf 1 auf Clustern mit weniger als vier Knoten kann zu einem HDFS-Datenverlust führen, wenn ein einzelner Knoten ausfällt. Wir empfehlen, für Produktionsworkloads einen Cluster mit mindestens vier Core-Knoten zu verwenden.
2. Amazon EMR erlaubt Clustern nicht, Core-Knoten unter `dfs.replication` zu skalieren. Bei `dfs.replication = 2` z. B. beträgt die Mindestanzahl von Core-Knoten 2.
3. Wenn Sie verwaltete Skalierung oder Auto-Scaling verwenden oder die Größe Ihres Clusters manuell ändern möchten, empfehlen wir Ihnen, `dfs.replication` auf 2 oder höher einzustellen.

Überlegungen zum Einrichten von Alarmen für Metriken:

- Amazon EMR stellt zurzeit keine anwendungsspezifischen Metriken zu HDFS oder YARN bereit. Wir empfehlen, dass Sie Alarme einrichten, um die Anzahl der primären Knoteninstanzen zu überwachen. Konfigurieren Sie die Alarme anhand der folgenden CloudWatch Amazon-Metriken: `MultiMasterInstanceGroupNodesRunning`, `MultiMasterInstanceGroupNodesRunningPercentage` oder `MultiMasterInstanceGroupNodesRequested`. CloudWatch benachrichtigt Sie, falls der Primärknoten ausfällt oder ausgetauscht wird.
- Wenn der niedriger als 100% und größer als 50% `MultiMasterInstanceGroupNodesRunningPercentage` ist, hat der Cluster möglicherweise einen Primärknoten verloren. In diesem Fall versucht Amazon EMR, einen Primärknoten zu ersetzen.
- Wenn der `MultiMasterInstanceGroupNodesRunningPercentage` Wert unter 50% fällt, sind möglicherweise zwei Primärknoten ausgefallen. In diesem Fall ist das Quorum verloren und der Cluster kann nicht wiederhergestellt werden. Sie müssen Daten manuell aus diesem Cluster migrieren.

Weitere Informationen finden Sie unter [Einrichten von Alarmen für Metriken](#).

EMR-Cluster auf AWS Outposts

Ab Amazon EMR 5.28.0 können Sie EMR-Cluster erstellen und ausführen. AWS Outposts ermöglicht native AWS Dienste, Infrastrukturen und Betriebsmodelle in lokalen

Einrichtungen. In AWS Outposts Umgebungen können Sie dieselben AWS APIs Tools und dieselbe Infrastruktur verwenden wie in der AWS Cloud. Amazon EMR on AWS Outposts ist ideal für Workloads mit niedriger Latenz, die in unmittelbarer Nähe zu lokalen Daten und Anwendungen ausgeführt werden müssen. [Weitere Informationen zu finden Sie im AWS Outposts Benutzerhandbuch.AWS Outposts](#)

Voraussetzungen

Im Folgenden sind die Voraussetzungen für die Verwendung von Amazon EMR in AWS Outposts aufgeführt:

- Sie müssen AWS Outposts in Ihrem lokalen Rechenzentrum installiert und konfiguriert haben.
- Sie müssen über eine zuverlässige Netzwerkverbindung zwischen Ihrer Outpost-Umgebung und einer AWS Region verfügen.
- Sie müssen über ausreichende Kapazität für Amazon EMR-unterstützte Instance-Typen in Ihrem Outpost verfügen.

Einschränkungen

Im Folgenden sind die Einschränkungen für die Verwendung von Amazon EMR auf AWS Outposts aufgeführt:

- On-Demand-Instances sind die einzige unterstützte Option für EC2 Amazon-Instances. Spot Instances sind für Amazon EMR auf AWS Outposts nicht verfügbar.
- Wenn Sie zusätzliche Amazon EBS-Speichervolumes benötigen, wird nur Allzweck-SSD (GP2) unterstützt.
- Wenn Sie die Amazon EMR-Versionen 5.28 bis 6.x verwenden AWS Outposts , können Sie nur S3-Buckets verwenden, die Objekte in einem AWS-Region von Ihnen angegebenen speichern. Mit Amazon EMR 7.0.0 und höher AWS Outposts wird Amazon EMR on auch mit dem S3A Dateisystem-Client, Präfix, unterstützt. `s3a://`
- Nur die folgenden Instance-Typen werden von Amazon EMR auf AWS Outposts unterstützt:

Instance-Klasse	Instance-Typen
Allgemeine Zwecke	m5.xlarge m5.2xlarge m5.4xlarge m5.12xlarge m5.24xlarge m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.12xlarge m5d.24xlarge
Für Datenverarbeitung optimiert	c5.xlarge c5.2xlarge c5.4xlarge c5.18xlarge c5d.xlarge c5d.2xlarge c5d.4xlarge c5d.18xlarge
RAM-optimiert	r5.xlarge r5.2xlarge r5.4xlarge r5.12xlarge r5d.xlarge r5d.2xlarge r5d.4xlarge r5d.12xlarge r5d.24xlarge
Speicheroptimiert	i3en.xlarge i3en.2xlarge i3en.3xlarge i3en.6xlarge i3en.12xlarge i3en.24xlarge

Überlegungen zur Netzwerkkonnektivität

- Wenn die Netzwerkverbindung zwischen Ihrem Outpost und seiner AWS Region unterbrochen wird, laufen Ihre Cluster weiter. Sie können jedoch keine neuen Cluster erstellen oder neue Aktionen für vorhandene Cluster ausführen, bis die Verbindung wiederhergestellt wurde. Bei Instance-Fehlern wird die Instance nicht automatisch ersetzt. Darüber hinaus werden Aktionen wie das Hinzufügen von Schritten zu einem laufenden Cluster, das Überprüfen des Ausführungsstatus der Schritte und das Senden von CloudWatch Metriken und Ereignissen verzögert.
- Wir empfehlen Ihnen, eine zuverlässige und hochverfügbare Netzwerkkonnektivität zwischen Ihrem Outpost und der AWS Region bereitzustellen. Wenn die Netzwerkverbindung zwischen Ihrem Outpost und seiner AWS Region für mehr als ein paar Stunden unterbrochen wird, laufen Cluster, für die der Terminierungsschutz aktiviert ist, weiter, und Cluster, die den Terminierungsschutz deaktiviert haben, können beendet werden.
- Falls die Netzwerkkonnektivität aufgrund einer routinemäßigen Wartung beeinträchtigt wird, empfehlen wir die proaktive Aktivierung des Beendigungsschutzes. Generell bedeutet die Unterbrechung der Konnektivität, dass externe Abhängigkeiten, die nicht lokal im Outpost oder

Kundennetzwerk sind, nicht zugänglich sind. Dazu gehören Amazon S3, DynamoDB, das mit EMRFS Consistency View verwendet wird, und Amazon RDS, wenn eine Instance in der Region für einen Amazon-EMR-Cluster mit mehreren Primärknoten verwendet wird.

Erstellen eines Amazon EMR-Clusters auf AWS Outposts

Das Erstellen eines Amazon EMR-Clusters auf AWS Outposts ähnelt dem Erstellen eines Amazon EMR-Clusters in der AWS Cloud. Wenn Sie einen Amazon EMR-Cluster auf erstellen AWS Outposts, müssen Sie ein EC2 Amazon-Subnetz angeben, das Ihrem Outpost zugeordnet ist.

Eine Amazon VPC kann sich über alle Availability Zones in einer AWS Region erstrecken. AWS Outposts sind Erweiterungen von Availability Zones, und Sie können eine Amazon VPC in einem Konto so erweitern, dass sie sich über mehrere Availability Zones und zugehörige Outpost-Standorte erstreckt. Wenn Sie den Outpost konfigurieren, ordnen Sie ihm ein Subnetz zu, um Ihre regionale VPC-Umgebung auf Ihre On-Premises-Einrichtung zu erweitern. Outpost-Instances und verwandte Services werden als Teil Ihrer regionalen VPC angezeigt, ähnlich einer Availability Zone mit verknüpften Subnetzen. Weitere Informationen finden Sie im [AWS Outposts -Benutzerhandbuch](#).

Konsole

Um einen neuen Amazon EMR-Cluster AWS Outposts mit dem zu erstellen AWS Management Console, geben Sie ein EC2 Amazon-Subnetz an, das mit Ihrem Outpost verknüpft ist.

Console

Um einen Cluster mit der Konsole zu AWS Outposts erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Clusterkonfiguration die Option Instance-Gruppen oder Instance-Flotten aus. Wählen Sie dann im Dropdownmenü Instanztyp auswählen einen EC2 Instance-Typ aus oder wählen Sie Aktionen und anschließend EBS-Volumes hinzufügen aus. Amazon EMR on AWS Outposts unterstützt begrenzte Amazon EBS-Volumes und Instance-Typen.
4. Wählen Sie unter Netzwerk ein EC2 Subnetz mit einer Outpost-ID in diesem Format aus: op-123456789.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.

- Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um einen Cluster zu erstellen mit dem AWS Outposts AWS CLI

- Um einen neuen Amazon EMR-Cluster AWS Outposts mit dem zu erstellen AWS CLI, geben Sie ein EC2 Subnetz an, das Ihrem Outpost zugeordnet ist, wie im folgenden Beispiel.
subnet-22XXXX01 Ersetzen Sie es durch Ihre eigene EC2 Amazon-Subnetz-ID.

```
aws emr create-cluster \
--name "Outpost cluster" \
--release-label emr-7.9.0 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey SubnetId=subnet-22XXXX01 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

EMR-Cluster in AWS Local Zones

Ab Amazon EMR Version 5.28.0 können Sie Amazon EMR-Cluster in einem Local Zones-Subnetz als logische Erweiterung einer AWS Region, die AWS Local Zones unterstützt, erstellen und ausführen. Eine lokale Zone ermöglicht es, Amazon EMR-Funktionen und eine Untergruppe von AWS Diensten, wie Rechen- und Speicherdienste, näher an den Benutzern zu platzieren, um Zugriff mit sehr geringer Latenz auf lokal ausgeführte Anwendungen zu ermöglichen. Eine Liste der verfügbaren Local Zones finden Sie unter [AWS Local Zones](#). Informationen zum Zugriff auf verfügbare AWS Local Zones finden Sie unter [Regionen, Availability Zones und lokale Zonen](#).

Unterstützte Instance-Typen

Die folgenden Instance-Typen sind für Amazon-EMR-Cluster auf Local Zones verfügbar. Die Verfügbarkeit des Instance-Typs kann je nach Region variieren.

Instance-Klasse	Instance-Typen
Allgemeine Zwecke	m5.xlarge m5.2xlarge m5.4xlarge m5.12xlarge m5.24xlarge m5d.xlarge m5d.2xlarge m5d.4xlarge m5d.12xlarge m5d.24xlarge

Instance-Klasse	Instance-Typen
Für Datenverarbeitung optimiert	c5.xlarge c5.2xlarge c5.4xlarge c5.9xlarge c5.18xlarge c5d.xlarge c5d.2xlarge c5d.4xlarge c5d.9xlarge c5d.18xlarge
RAM-optimiert	r5.xlarge r5.2xlarge r5.4xlarge r5.12xlarge r5d.xlarge r5d.2xlarge r5d.4xlarge r5d.12xlarge r5d.24xlarge
Speicheroptimiert	i3en.xlarge i3en.2xlarge i3en.3xlarge i3en.6xlarge i3en.12xlarge i3en.24xlarge

Erstellen eines Amazon-EMR-Clusters auf Local Zones

Erstellen Sie einen Amazon EMR-Cluster in AWS Local Zones, indem Sie den Amazon EMR-Cluster in einem Amazon VPC-Subnetz starten, das mit einer lokalen Zone verknüpft ist. Sie können auf den Cluster mit dem Local-Zone-Namen zugreifen, z. B. us-west-2-lax-1a in der USA West (Oregon)-Konsole.

Local Zones unterstützen derzeit keine Amazon EMR Notebooks oder direkte Verbindungen zu Amazon EMR über die Schnittstelle VPC-Endpunkt ().AWS PrivateLink

Console

Um mit der Konsole einen Cluster in einer lokalen Zone zu erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Netzwerk ein EC2 Subnetz mit einer lokalen Zonen-ID in diesem Format aus: subnet 123abc | us-west-2-lax-1a.
4. Wählen Sie den Instance-Typ oder fügen Sie Amazon-EBS-Speichervolumes für einheitliche Instance-Gruppen oder Instance-Flotten hinzu.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um einen Cluster in einer lokalen Zone mit dem zu erstellen AWS CLI

- Verwenden Sie den Befehl `create-cluster` zusammen mit dem Befehl `SubnetId` for the Local Zone, wie im folgenden Beispiel gezeigt. Ersetzen Sie `subnet-22 XXXX1234567` durch die Lokale Zone `SubnetId` und ersetzen Sie gegebenenfalls andere Optionen. Weitere Informationen finden Sie unter <https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html>.

```
aws emr create-cluster \  
--name "Local Zones cluster" \  
--release-label emr-5.29.0 \  
--applications Name=Spark \  
--ec2-attributes KeyName=myKey,SubnetId=subnet-22XXXX1234567 \  
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Docker für die Verwendung mit Amazon EMR-Clustern konfigurieren

Amazon EMR 6.x unterstützt Hadoop 3, wodurch YARN NodeManager Container entweder direkt auf dem Amazon EMR-Cluster oder in einem Docker-Container starten kann. Docker-Container bieten benutzerdefinierte Ausführungsumgebungen, in denen Anwendungscode ausgeführt wird. Die benutzerdefinierte Ausführungsumgebung ist von der Ausführungsumgebung von YARN und anderen Anwendungen isoliert. NodeManager

Docker-Container können spezielle Bibliotheken enthalten, die von der Anwendung verwendet werden, und sie können verschiedene Versionen von systemeigenen Tools und Bibliotheken wie R und Python bereitstellen. Sie können die vertrauten Docker-Tools verwenden, um Bibliotheken und Laufzeitabhängigkeiten für Ihre Anwendungen zu definieren.

Amazon-EMR-6.x-Cluster sind standardmäßig so konfiguriert, dass YARN Anwendungen wie Spark mit Docker-Containern ausführen kann. Um die Containerkonfiguration anzupassen, bearbeiten Sie die Docker-Unterstützungsoptionen, die in den im `/etc/hadoop/conf`-Verzeichnis verfügbaren Dateien `„yarn-site.xml“` und `„container-executor.cfg“` definiert sind. Weitere Informationen zu den einzelnen Konfigurationsoptionen und deren Verwendung finden Sie unter [Starten von Anwendungen mithilfe von Docker-Containern](#).

Sie können Docker verwenden, wenn Sie eine Aufgabe absenden. Verwenden Sie die folgenden Variablen, um die Docker-Laufzeit und das Docker-Image anzugeben.

- `YARN_CONTAINER_RUNTIME_TYPE=docker`
- `YARN_CONTAINER_RUNTIME_DOCKER_IMAGE={DOCKER_IMAGE_NAME}`

Wenn Sie Docker-Container verwenden, um Ihre YARN-Anwendungen auszuführen, lädt YARN das Docker-Image herunter, das Sie beim Absenden der Aufgabe angeben. Damit YARN dieses Docker-Image auflösen kann, muss es mit einer Docker-Registrierung konfiguriert werden. Die Konfigurationsoptionen für eine Docker-Registrierung hängen davon ab, ob Sie den Cluster über ein öffentliches oder privates Subnetz bereitstellen.

Docker-Registrierungen

Eine Docker-Registrierung ist ein Speicher- und Verteilungssystem für Docker-Images. Für Amazon EMR empfehlen wir die Verwendung von Amazon ECR, einer vollständig verwalteten Docker-Container-Registrierung, mit der Sie Ihre eigenen benutzerdefinierten Images erstellen und diese in einer hochverfügbaren und skalierbaren Architektur hosten können.

Überlegungen zur Bereitstellung

Für Docker-Registrierungen ist Netzwerkzugriff von jedem Host im Cluster erforderlich. Dies liegt daran, dass jeder Host Images aus der Docker-Registrierung herunterlädt, wenn Ihre YARN-Anwendung auf dem Cluster ausgeführt wird. Diese Anforderungen an die Netzwerkkonnektivität können die Auswahl der Docker-Registrierung einschränken, je nachdem, ob Sie Ihren Amazon-EMR-Cluster in einem öffentlichen oder privaten Subnetz bereitstellen.

Public subnet (Öffentliches Subnetz)

Wenn EMR-Cluster in einem öffentlichen Subnetz bereitgestellt werden, NodeManager können die Knoten, auf denen YARN ausgeführt wird, direkt auf jede Registrierung zugreifen, die über das Internet verfügbar ist.

Privates Subnetz

Wenn EMR-Cluster in einem privaten Subnetz bereitgestellt werden, haben die Knoten, auf denen YARN ausgeführt wird, NodeManager keinen direkten Zugriff auf das Internet. Docker-Images können in Amazon ECR gehostet und über abgerufen werden. AWS PrivateLink

Weitere Informationen dazu, wie Sie AWS PrivateLink den Zugriff auf Amazon ECR in einem privaten Subnetzscenario zulassen, finden Sie unter [Einrichtung AWS PrivateLink für Amazon ECS und Amazon ECR](#).

Konfigurieren von Docker-Registrierungen

Um Docker-Registrierungen mit Amazon EMR verwenden zu können, müssen Sie Docker so konfigurieren, dass sie der spezifischen Registrierung vertrauen, die Sie zum Auflösen von Docker-Images verwenden möchten. Die Standardvertrauensregistrierungen sind „local“ (privat) und „centos“. Wenn Sie andere öffentliche Repositorys oder Amazon ECR verwenden möchten, können Sie `docker.trusted.registries`-Einstellungen in `/etc/hadoop/conf/container-executor.cfg` außer Kraft setzen, indem Sie die EMR-Klassifikations-API mit dem `container-executor`-Klassifizierungsschlüssel verwenden.

Das folgende Beispiel zeigt, wie Sie den Cluster so konfigurieren, dass er sowohl einem öffentlichen Repository mit dem Namen „your-public-repo“, als auch einem ECR-Registrierungsendpunkt vertraut, `123456789123.dkr.ecr.us-east-1.amazonaws.com`. Wenn Sie ECR verwenden, ersetzen Sie diesen Endpunkt durch Ihren spezifischen ECR-Endpunkt.

```
[
  {
    "Classification": "container-executor",
    "Configurations": [
      {
        "Classification": "docker",
        "Properties": {
          "docker.trusted.registries": "local,centos,your-public-repo,123456789123.dkr.ecr.us-east-1.amazonaws.com",
          "docker.privileged-containers.registries": "local,centos,your-public-repo,123456789123.dkr.ecr.us-east-1.amazonaws.com"
        }
      }
    ]
  }
]
```

Um einen Amazon EMR 6.0.0-Cluster mit dieser Konfiguration mithilfe von AWS Command Line Interface (AWS CLI) zu starten, erstellen Sie eine Datei `container-executor.json` mit dem Namen des Inhalts der vorherigen Container-Executor-JSON-Konfiguration. Verwenden Sie dann die folgenden Befehle, um den Cluster zu starten.


```
export KEYPAIR=<Name of your Amazon EC2 key-pair>
export SUBNET_ID=<ID of the subnet to which to deploy the cluster>
export INSTANCE_TYPE=<Name of the instance type to use>
export REGION=<Region to which to deploy the cluster>

aws emr create-cluster \
  --name "EMR-6.0.0" \
  --region $REGION \
  --release-label emr-6.0.0 \
  --applications Name=Hadoop Name=Spark \
  --service-role EMR_DefaultRole \
  --ec2-attributes KeyName=$KEYPAIR,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=
$SUBNET_ID \
  --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=
$INSTANCE_TYPE InstanceGroupType=CORE,InstanceCount=2,InstanceType=$INSTANCE_TYPE \
  --configuration file://container-executor.json
```

YARN für den Zugriff auf Amazon ECR auf EMR 6.0.0 und früher konfigurieren

Wenn Sie neu bei Amazon ECR sind, folgen Sie den Anweisungen unter [Erste Schritte mit Amazon ECR](#) und stellen Sie sicher, dass Sie von jeder Instance in Ihrem Amazon-EMR-Cluster Zugriff auf Amazon ECR haben.

Um unter EMR 6.0.0 und früher auf Amazon ECR mit dem Docker-Befehl zuzugreifen, müssen Sie zunächst Anmeldeinformationen generieren. Wenn Sie überprüfen möchten, ob YARN auf Images von Amazon ECR zugreifen kann, verwenden Sie die Container-Umgebungsvariable `YARN_CONTAINER_RUNTIME_DOCKER_CLIENT_CONFIG`, um einen Verweis auf die von Ihnen generierten Anmeldeinformationen zu übergeben.

Führen Sie den folgenden Befehl auf einem der Core-Knoten aus, um die Anmeldezeile für Ihr ECR-Konto zu erhalten.

```
aws ecr get-login --region us-east-1 --no-include-email
```

Der `get-login`-Befehl generiert den korrekten Docker-CLI-Befehl zum Erstellen von Anmeldeinformationen. Kopieren Sie die Ausgabe von `get-login` und führen Sie sie aus.

```
sudo docker login -u AWS -p <password> https://<account-id>.dkr.ecr.us-east-1.amazonaws.com
```

Mit diesem Befehl wird eine `config.json`-Datei im `/root/.docker`-Ordner generiert. Kopieren Sie diese Datei in HDFS, damit sie von den an den Cluster übermittelten Aufgaben zur Authentifizierung bei Amazon ECR verwendet werden kann.

Führen Sie die folgenden Befehle aus, um die `config.json`-Datei in Ihr Startverzeichnis zu kopieren.

```
mkdir -p ~/.docker
sudo cp /root/.docker/config.json ~/.docker/config.json
sudo chmod 644 ~/.docker/config.json
```

Führen Sie die folgenden Befehle aus, um die `config.json`-Datei in HDFS festzulegen, damit sie von Aufgaben verwendet werden kann, die auf dem Cluster ausgeführt werden.

```
hadoop fs -put ~/.docker/config.json /user/hadoop/
```

YARN kann auf ECR als Docker-Image-Registrierung zugreifen und Container während der Aufgabenausführung abrufen.

Nachdem Sie Docker-Registrierungen und YARN konfiguriert haben, können Sie YARN-Anwendungen mit Docker-Containern ausführen. Weitere Informationen finden Sie unter [Ausführen von Spark-Anwendungen mit Docker mithilfe von Amazon EMR 6.0.0](#).

In EMR 6.1.0 und höher müssen Sie die Authentifizierung für Amazon ECR nicht manuell einrichten. Wenn im `container-executor`-Klassifikationsschlüssel eine Amazon-ECR-Registrierung erkannt wird, wird das automatische Amazon-ECR-Authentifizierungsfeature aktiviert, und YARN wickelt den Authentifizierungsprozess ab, wenn Sie einen Spark-Auftrag mit einem ECR-Image einreichen. Sie können überprüfen, ob die automatische Authentifizierung aktiviert ist, indem Sie in `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` der YARN-Seite einchecken. Die automatische Authentifizierung ist aktiviert und die YARN-Authentifizierungseinstellung ist auf `true` gesetzt, wenn die eine ECR-Registrierungs-URL `docker.trusted.registries` enthält.

Voraussetzungen für die Verwendung der automatischen Authentifizierung bei Amazon ECR

- EMR Version 6.1.0 oder höher
- Die in der Konfiguration enthaltene ECR-Registrierung befindet sich in derselben Region wie der Cluster

- IAM-Rolle mit Berechtigungen zum Abrufen des Autorisierungstoken und zum Abrufen eines beliebigen Images

Weitere Informationen finden Sie unter [Einrichten mit Amazon ECS](#).

Wie aktiviere ich die automatische Authentifizierung

Folgen Sie [Konfigurieren von Docker-Registrierungen](#), um eine Amazon-ECR-Registrierung als vertrauenswürdige Registrierung festzulegen, und stellen Sie sicher, dass sich das Amazon-ECR-Repository und der Cluster in derselben Region befinden.

Um dieses Feature auch dann zu aktivieren, wenn die ECR-Registrierung nicht in der vertrauenswürdigen Registrierung festgelegt ist, verwenden Sie die Konfigurationsklassifizierung, um `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` auf `true` festzulegen.

Wie deaktiviere ich die automatische Authentifizierung

Standardmäßig ist die automatische Authentifizierung deaktiviert, wenn in der vertrauenswürdigen Registrierung keine Amazon-ECR-Registrierung erkannt wird.

Um die automatische Authentifizierung zu deaktivieren, auch wenn die Amazon-ECR-Registrierung in der vertrauenswürdigen Registrierung festgelegt ist, verwenden Sie die Konfigurationsklassifizierung, um `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` auf `false` festzulegen.

Wie überprüft man, ob die automatische Authentifizierung in einem Cluster aktiviert ist

Verwenden Sie einen Text-Editor wie `vi` auf dem Hauptknoten, um den Inhalt der Datei `vi /etc/hadoop/conf.empty/yarn-site.xml` anzuzeigen. Überprüfen Sie den Wert von `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled`.

Steuern Sie die Kündigung des Amazon EMR-Clusters

In diesem Abschnitt werden Ihre Optionen zum Herunterfahren von Amazon-EMR-Clustern beschrieben. Es behandelt automatische Kündigung und Kündigungsschutz sowie deren Interaktion mit anderen Amazon-EMR-Features.

Sie können einen Amazon-EMR-Cluster folgendermaßen herunterfahren:

- Kündigung nach der Ausführung des letzten Schritts – Erstellen Sie einen vorübergehenden Cluster, der nach Abschluss aller Schritte heruntergefahren wird.
- Automatische Kündigung (nach Inaktivität) – Erstellen Sie einen Cluster mit einer automatischen Terminierungsrichtlinie, der nach einer bestimmten Leerlaufzeit heruntergefahren wird. Weitere Informationen finden Sie unter [Verwenden einer automatischen Terminierungsrichtlinie für die Amazon EMR-Clusterbereinigung](#).
- Manuelles Beenden – Erstellen Sie einen Cluster mit langer Laufzeit, der so lange läuft, bis Sie ihn bewusst beenden. Informationen zum manuellen Beenden eines Clusters finden Sie unter [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#).

Sie können auch einen Kündigungsschutz für einen Cluster einrichten, um zu verhindern, dass EC2 Instances versehentlich oder versehentlich heruntergefahren werden.

Wenn Amazon EMR Ihren Cluster herunterfährt, werden alle EC2 Amazon-Instances im Cluster heruntergefahren. Daten im Instance-Speicher und auf EBS-Volumes sind nicht mehr verfügbar und können nicht wiederhergestellt werden. Es ist von kritischer Bedeutung, das Beenden von Clustern zu verstehen und zu kontrollieren, um eine Strategie für die Verwaltung und Bewahrung von Daten erstellen zu können, bei der die Daten zu Amazon S3 geschrieben und die Kosten abgewogen werden.

Themen

- [Konfiguration eines Amazon EMR-Clusters zum Fortfahren oder Beenden nach der Ausführung des Schritts](#)
- [Verwenden einer automatischen Terminierungsrichtlinie für die Amazon EMR-Clusterbereinigung](#)
- [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#)

Konfiguration eines Amazon EMR-Clusters zum Fortfahren oder Beenden nach der Ausführung des Schritts

In diesem Thema werden die Unterschiede zwischen der Verwendung eines Clusters mit langer Laufzeit und der Erstellung eines transienten Clusters erläutert, der nach der Ausführung des letzten Schritts heruntergefahren wird. Außerdem wird beschrieben, wie die Schrittausführung für einen Cluster konfiguriert wird.

So erstellen Sie einen langlebigen Cluster

Standardmäßig laufen Cluster, die Sie mit der Konsole oder der AWS CLI erstellen, lange. Cluster mit langer Laufzeit laufen weiter, akzeptieren Arbeit und es fallen Gebühren an, bis Sie Maßnahmen ergreifen, um sie herunterzufahren.

Ein Cluster mit langer Laufzeit ist in folgenden Situationen wirksam:

- Wenn Sie interaktiv oder automatisch Daten abfragen müssen.
- Wenn Sie kontinuierlich mit Big-Data-Anwendungen interagieren müssen, die auf dem Cluster gehostet werden.
- Wenn Sie regelmäßig einen Datensatz verarbeiten, der so groß oder so häufig ist, dass es ineffizient ist, jedes Mal neue Cluster zu starten und Daten zu laden.

Sie können auch einen Kündigungsschutz für einen Cluster mit langer Laufzeit einrichten, um zu verhindern, dass EC2 Instanzen versehentlich oder versehentlich heruntergefahren werden. Weitere Informationen finden Sie unter [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).

Note

Amazon EMR aktiviert automatisch den Kündigungsschutz für alle Cluster mit mehreren Primärknoten und überschreibt alle Einstellungen für die Schrittausführung, die Sie bei der Erstellung des Clusters angeben. Sie können den Kündigungsschutz deaktivieren, nachdem der Cluster gestartet wurde. Siehe [Konfigurieren des Beendigungsschutzes für aktive Cluster](#). Um einen Cluster mit mehreren Primärknoten herunterzufahren, müssen Sie zunächst die Clusterattribute ändern, um den Kündigungsschutz zu deaktivieren. Detaillierte Anweisungen finden Sie unter [Amazon-EMR-Clustern mit mehreren Primärknoten beenden](#).

Einen Cluster so konfigurieren, dass er nach der Ausführung des Schritts beendet wird

Wenn Sie die Beendigung nach der Schrittausführung konfigurieren, startet der Cluster, führt Bootstrap-Aktionen aus und führt dann die von Ihnen angegebenen Schritte aus. Sobald der letzte Schritt abgeschlossen ist, beendet Amazon EMR die EC2 Amazon-Instances des Clusters. Bei Clustern, die Sie mit der Amazon-EMR-API starten, ist die Schrittausführung standardmäßig aktiviert.

Die Beendigung nach der Schrittausführung ist für Cluster wirksam, die eine periodische Verarbeitungsaufgabe ausführen, beispielsweise einen täglichen Datenverarbeitungslauf. Mit der schrittweisen Ausführung können Sie außerdem sicherstellen, dass Ihnen nur die Zeit in Rechnung gestellt wird, die für die Verarbeitung Ihrer Daten erforderlich ist. Weitere Informationen zu den Schritten finden Sie unter [Arbeit an einen Amazon EMR-Cluster einreichen](#).

Console

Um die Beendigung nach der Ausführung des Schritts mit der Konsole zu aktivieren

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Schritte die Option Schritt hinzufügen aus. Geben Sie im Dialogfeld Schritt hinzufügen die entsprechenden Feldwerte ein. Die Optionen unterscheiden sich je nach Schritttyp. Um Ihren Schritt hinzuzufügen und das Dialogfeld zu verlassen, wählen Sie Schritt hinzufügen.
4. Aktivieren Sie unter Clusterbeendigung das Kontrollkästchen Cluster nach Abschluss des letzten Schritts beenden.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um die Beendigung nach der Ausführung des Schritts zu aktivieren, verwenden Sie AWS CLI

- Geben Sie den `--auto-terminate`-Parameter an, wenn Sie den `create-cluster`-Befehl verwenden, um einen vorübergehenden Cluster zu erstellen.

Das folgende Beispiel veranschaulicht die Verwendung des `--auto-terminate`-Parameters. Sie können den folgenden Befehl eingeben und ihn *myKey* durch den Namen Ihres EC2 key pair ersetzen.

 Note

Linux-Zeilenfortsetzungszeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.9.0 \
--applications Name=Hive Name=Pig --use-default-roles --ec2-attributes
KeyName=myKey \
--steps Type=PIG,Name="Pig Program",ActionOnFailure=CONTINUE,\
Args=[-f,s3://amzn-s3-demo-bucket/scripts/pigscript.pig,-p,\
INPUT=s3://amzn-s3-demo-bucket/inputdata/,-p,OUTPUT=s3://amzn-s3-demo-bucket/
outputdata/,\
$INPUT=s3://amzn-s3-demo-bucket/inputdata/,$OUTPUT=s3://amzn-s3-demo-bucket/
outputdata/]
--instance-type m5.xlarge --instance-count 3 --auto-terminate
```

API

So deaktivieren Sie die Beendigung nach der Schrittausführung mit der Amazon EMR-API beim Clusterstart

1. Wenn Sie die [RunJobFlow](#)Aktion verwenden, um einen Cluster zu erstellen, setzen Sie die [KeepJobFlowAliveWhenNoSteps](#)Eigenschaft auf `false`.
2. So ändern Sie Ihre Konfiguration der Beendigung nach der Ausführung des Schritts mit der Amazon EMR-API nach dem Start des Clusters:

SetKeepJobFlowAliveWhenNoSteps Aktion verwenden.

Verwenden einer automatischen Terminierungsrichtlinie für die Amazon EMR-Clusterbereinigung

Mit einer Richtlinie zur automatischen Terminierung können Sie die Clusterbereinigung orchestrieren, ohne ungenutzte Cluster überwachen und manuell beenden zu müssen. Wenn Sie einem Cluster

eine automatische Terminierungsrichtlinie hinzufügen, geben Sie die Leerlaufzeit an, nach der der Cluster automatisch heruntergefahren werden soll.

Je nach Release-Version verwendet Amazon EMR unterschiedliche Kriterien, um einen Cluster als inaktiv zu kennzeichnen. In der folgenden Tabelle wird beschrieben, wie Amazon EMR den Cluster-Leerlauf bestimmt.

Wenn Sie ...	Ein Cluster gilt als inaktiv, wenn ...
Amazon-EMR-Versionen 5.34.0 und höher und 6.4.0 und höher	• Es gibt keine aktiven YARN-Anwendungen • Die HDFS-Auslastung liegt unter 10 % • Es gibt keine aktiven EMR-Notebook- oder EMR-Studio-Verbindungen • Es werden keine Benutzeroberflächen für Cluster-Anwendungen verwendet • Es gibt keine ausstehenden Schritte
Amazon-EMR-Versionen 5.30.0 – 5.33.0 und 6.1.0 – 6.3.0	• Es gibt keine aktiven YARN-Anwendungen • Der Cluster hat keine aktiven Spark-Aufträge  Note Amazon EMR markiert einen Cluster als inaktiv und beendet den Cluster möglicherweise automatisch, auch wenn Sie über einen aktiven Python3-Kernel verfügen. Das liegt daran, dass bei der Ausführung eines Python3-Kernels kein Spark-Job auf dem Cluster gesendet wird. Um die automatische Terminierung mit einem Python3-K

Wenn Sie ...	Ein Cluster gilt als inaktiv, wenn ...
	ernel zu verwenden, empfehlen wir die Verwendung von Amazon-EMR-Version 6.4.0 oder höher.

Note

Amazon-EMR-Versionen 6.4.0 und höher unterstützen eine Cluster-Datei zur Erkennung von Aktivitäten auf dem Primärknoten: `/emr/metricscollector/isbusy`. Wenn Sie einen Cluster verwenden, um Shell-Skripts oder Nicht-YARN-Anwendungen auszuführen, können Sie Amazon EMR regelmäßig bearbeiten oder `isbusy` aktualisieren, um Amazon EMR mitzuteilen, dass sich der Cluster nicht im Leerlauf befindet.

Sie können beim Erstellen eines Clusters eine automatische Terminierungsrichtlinie anhängen oder einem vorhandenen Cluster eine Richtlinie hinzufügen. Um die automatische Kündigung zu ändern oder zu deaktivieren, können Sie die Richtlinie aktualisieren oder entfernen.

Überlegungen

Berücksichtigen Sie die folgenden Features und Einschränkungen, bevor Sie eine Richtlinie zum automatischen Beenden verwenden:

- Im Folgenden AWS-Regionen ist die automatische Kündigung von Amazon EMR mit Amazon EMR 6.14.0 und höher verfügbar:
 - Europa (Spanien) (eu-south-2)
- Im Folgenden AWS-Regionen ist die automatische Kündigung von Amazon EMR mit Amazon EMR 5.30.0 und 6.1.0 und höher verfügbar:
 - USA Ost (Nord-Virginia): (us-east-1)
 - USA Ost (Ohio): (us-east-2)
 - USA West (Oregon): (us-west-2)
 - USA West (Nordkalifornien) (us-west-1)
 - Afrika (Kapstadt) (af-south-1)
 - Asien-Pazifik (Hongkong) (ap-east-1)

- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Hyderabad) (ap-south-2)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Osaka) (ap-northeast-3)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Jakarta) (ap-southeast-3)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Kanada (Zentral): (ca-central-1)
- Südamerika (São Paulo) (sa-east-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Zürich) (eu-central-2)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Mailand) (eu-south-1)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Israel (Tel Aviv) (il-central-1)
- Naher Osten (VAE) (me-central-1)
- China (Peking) (cn-north-1)
- China (Ningxia) (cn-northwest-1)
- AWS GovCloud (US-Ost) (-1) us-gov-east
- AWS GovCloud (US-West) (us-gov-west-1)
- Das Leerlauf-Timeout ist standardmäßig auf 60 Minuten (eine Stunde) eingestellt, wenn Sie keinen Wert angeben. Sie können ein minimales Timeout für den Leerlauf von einer Minute und ein maximales Timeout für den Leerlauf von 7 Tagen angeben.
- Bei Amazon-EMR-Versionen 6.4.0 und höher ist die automatische Terminierung standardmäßig aktiviert, wenn Sie mit der Amazon-EMR-Konsole einen neuen Cluster erstellen.
- Amazon EMR veröffentlicht hochauflösende Amazon CloudWatch Metriken, wenn Sie die automatische Terminierung für einen Cluster aktivieren. Sie können diese Metriken verwenden,

um Cluster-Aktivität und Inaktivität zu verfolgen. Weitere Informationen finden Sie unter [Cluster-Kapazitätsmetriken](#).

- Die automatische Terminierung wird nicht unterstützt, wenn Sie Anwendungen verwenden, die nicht auf Yarn basieren, wie Presto, Trino oder. HBase
- Um die automatische Terminierung zu verwenden, muss der Metrics-Collector-Prozess in der Lage sein, eine Verbindung zum öffentlichen API-Endpunkt für die automatische Terminierung in API Gateway herzustellen. Wenn Sie einen privaten DNS-Namen mit verwenden Amazon Virtual Private Cloud, funktioniert die automatische Terminierung nicht ordnungsgemäß. Um sicherzustellen, dass die automatische Beendigung funktioniert, empfehlen wir Ihnen, eine der folgenden Maßnahmen zu ergreifen:
 - Entfernen Sie den VPC-Endpunkt der API-Gateway-Schnittstelle aus Ihrer Amazon VPC.
 - Folgen Sie den Anweisungen unter [Warum erhalte ich den Fehler HTTP 403 Forbidden, wenn ich von einer VPC APIs aus eine Verbindung zu meinem API Gateway herstelle?](#) um die Einstellung für den privaten DNS-Namen zu deaktivieren.
 - Starten Sie Ihren Cluster stattdessen in einem privaten Subnetz. Weitere Informationen finden Sie im Thema [Private Subnetze](#).
- (EMR 5.30.0 und höher) Wenn Sie die Standardregel Allow All Outbound für die primäre Sicherheitsgruppe auf 0.0.0.0/ entfernen, müssen Sie Ihrer Sicherheitsgruppe eine Regel hinzufügen, die ausgehende TCP-Konnektivität für den Servicezugriff auf Port 9443 zulässt. Die Sicherheitsgruppe für den Servicezugriff muss eingehenden Datenverkehr über TCP- und UDP-Port 53 von Ihrer primären Sicherheitsgruppe zulassen. Weitere Informationen über die Konfiguration von Sicherheitsgruppen finden Sie unter Von [Amazon EMR verwaltete Sicherheitsgruppe für die primäre Instance \(private Subnetze\)](#).

Berechtigungen zur Verwendung der automatischen Beendigung

Bevor Sie Richtlinien zur automatischen Kündigung für Amazon EMR anwenden und verwalten können, müssen Sie die in der folgenden Beispiel-IAM-Berechtigungsrichtlinie aufgeführten Berechtigungen den IAM-Ressourcen zuordnen, die Ihren EMR-Cluster verwalten.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AllowAutoTerminationPolicyActions",
    "Effect": "Allow",
    "Action": [
```

```
    "elasticmapreduce:PutAutoTerminationPolicy",
    "elasticmapreduce:GetAutoTerminationPolicy",
    "elasticmapreduce:RemoveAutoTerminationPolicy"
  ],
  "Resource": "<your-resources>"
}
```

Eine Richtlinie zur automatischen Beendigung anhängen, aktualisieren oder entfernen

Dieser Abschnitt enthält Anweisungen, die Ihnen helfen, eine Richtlinie zur automatischen Beendigung an einen Amazon-EMR-Cluster anzuhängen, zu aktualisieren oder zu entfernen. Bevor Sie mit Richtlinien zur automatischen Beendigung arbeiten, stellen Sie sicher, dass Sie über die erforderlichen IAM-Berechtigungen verfügen. Siehe [Berechtigungen zur Verwendung der automatischen Beendigung](#).

Console

Um eine automatische Terminierungsrichtlinie anzuhängen, wenn Sie einen Cluster mit der Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Clusterbeendigung die Option Cluster nach Leerlauf beenden aus.
4. Geben Sie die Anzahl der Stunden und Minuten im Leerlauf an, die vergehen können, bis der Cluster automatisch beendet wird. Die standardmäßige Leerlaufzeit beträgt eine Stunde.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

Um eine automatische Terminierungsrichtlinie auf einem laufenden Cluster mit der Konsole anzuhängen, zu aktualisieren oder zu entfernen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.

3. Suchen Sie auf der Registerkarte Eigenschaften der Cluster-Detailseite nach Clusterbeendigung und wählen Sie Bearbeiten aus.
4. Wählen oder deaktivieren Sie Automatische Beendigung aktivieren, um das Feature ein- oder auszuschalten. Wenn Sie die automatische Terminierung aktivieren, geben Sie die Anzahl der Stunden und Minuten im Leerlauf an, die vergehen können, bis der Cluster automatisch beendet wird. Wählen Sie dann zur Bestätigung Änderungen speichern aus.

AWS CLI

Bevor Sie beginnen

Bevor Sie mit Richtlinien zur automatischen Kündigung arbeiten, empfehlen wir Ihnen, auf die neueste Version von AWS CLI zu aktualisieren. Anweisungen finden Sie unter [Installieren, Aktualisieren und Deinstallieren von AWS CLI](#).

Um eine automatische Beendigungsrichtlinie anzuhängen oder zu aktualisieren, verwenden Sie AWS CLI

- Sie können den `aws emr put-auto-termination-policy`-Befehl verwenden, um eine automatische Beendigungsrichtlinie für einen Cluster anzuhängen oder zu aktualisieren.

Im folgenden Beispiel werden 3600 Sekunden für angegeben. *IdleTimeout* Wenn Sie nichts angeben *IdleTimeout*, ist der Standardwert eine Stunde.

```
aws emr put-auto-termination-policy \  
--cluster-id <your-cluster-id> \  
--auto-termination-policy IdleTimeout=3600
```

Note

Linux-Zeilenumbruchzeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

Sie können auch einen Wert für `--auto-termination-policy` angeben, wenn Sie den `aws emr create-cluster`-Befehl verwenden. Weitere Informationen zur Verwendung von Amazon EMR-Befehlen in der AWS CLI finden Sie in der [AWS CLI Befehlsreferenz](#).

Um eine automatische Kündigungsrichtlinie mit dem zu entfernen AWS CLI

- Verwenden Sie den `aws emr remove-auto-termination-policy`-Befehl, um eine automatische Beendigungsrichtlinie aus einem Cluster zu entfernen. Weitere Informationen zur Verwendung von Amazon EMR-Befehlen in der AWS CLI finden Sie in der [AWS CLI Befehlsreferenz](#).

```
aws emr remove-auto-termination-policy --cluster-id <your-cluster-id>
```

Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen

Der Terminierungsschutz schützt Ihre Cluster vor einer versehentlichen Kündigung. Dies kann besonders bei Clustern mit langer Laufzeit, die kritische Workloads verarbeiten, nützlich sein. Wenn der Beendigungsschutz für einen langlebigen Cluster aktiviert ist, können Sie den Cluster weiter beenden, müssen jedoch zunächst den Beendigungsschutz explizit aus dem Cluster entfernen. Dadurch wird sichergestellt, dass EC2 Instances nicht versehentlich oder irrtümlich heruntergefahren werden. Sie können den Beendigungsschutz aktivieren, wenn Sie einen Cluster erstellen. Sie können die Einstellung auf einem ausgeführten Cluster ändern.

Wenn der Beendigungsschutz aktiviert ist, funktioniert die Aktion `TerminateJobFlows` in der Amazon-EMR-API nicht. Benutzer können den Cluster nicht über diese API oder den Befehl `terminate-clusters` in der AWS CLI beenden. Die API gibt einen Fehler zurück und die CLI wird mit einem Rückgabecode ungleich null beendet. Wenn Sie die Amazon-EMR-Konsole verwenden, um einen Cluster zu beenden, werden Sie zu einem zusätzlichen Schritt aufgefordert, um den Beendigungsschutz zu deaktivieren.

Warning

Der Kündigungsschutz garantiert nicht, dass Daten im Falle eines menschlichen Fehlers oder einer Behelfslösung erhalten bleiben, z. B. wenn über die Befehlszeile ein Neustartbefehl ausgegeben wird, während eine SSH-Verbindung mit der Instance besteht, wenn eine Anwendung oder ein Skript, das auf der Instance ausgeführt wird, einen Neustartbefehl ausgibt oder wenn die Amazon- EC2 oder Amazon EMR-API verwendet wird, um den Kündigungsschutz zu deaktivieren. Dies gilt auch, wenn Sie Amazon EMR-Versionen 7.1 und höher ausführen und eine Instance fehlerhaft und nicht wiederherstellbar ist. Selbst wenn der Beendigungsschutz aktiviert ist, können im Instance-Speicher gespeicherte

Daten, einschließlich HDFS-Daten, verloren gehen. Schreiben Sie die Datenausgabe an Amazon-S3-Standorte und erstellen Sie Backup-Strategien, die Ihren Anforderungen an die Geschäftskontinuität entsprechen.

Der Beendigungsschutz wirkt sich nicht auf Ihre Fähigkeit aus, Cluster-Ressourcen mit einer der folgenden Aktionen zu skalieren:

- Manuelles Ändern der Größe eines Clusters mit dem oder der AWS Management Console AWS CLI. Weitere Informationen finden Sie unter [Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters](#).
- Entfernen von Instances aus einer Core- oder Aufgaben-Instance-Gruppe unter Verwendung einer Abwärtsskalierungsrichtlinie mit Auto Scaling. Weitere Informationen finden Sie unter [Verwenden der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instanzgruppen in Amazon EMR](#).
- Entfernen von Instances aus einer Instance-Flotte durch Reduzierung der Zielkapazität. Weitere Informationen finden Sie unter [Instance-Flotten-Optionen](#).

Kündigungsschutz und Amazon EC2

Die Einstellung für den Kündigungsschutz in einem Amazon EMR-Cluster entspricht dem `DisableApiTermination` Attribut für alle EC2 Amazon-Instances im Cluster. Wenn Sie beispielsweise den Kündigungsschutz in einem EMR-Cluster aktivieren, setzt Amazon EMR automatisch für alle EC2 Instances innerhalb des EMR-Clusters `DisableApiTermination` auf `true`. Das Gleiche gilt, wenn Sie den Kündigungsschutz deaktivieren. Amazon EMR wird für alle EC2 Instances innerhalb des EMR-Clusters automatisch auf `False` gesetzt. `DisableApiTermination` Wenn Sie einen Cluster von Amazon EMR aus beenden oder herunterskalieren und die EC2 Amazon-Einstellungen für eine EC2 Instance in Konflikt geraten, priorisiert Amazon EMR die Amazon EMR-Einstellung vor den `DisableApiTermination` Einstellungen `DisableApiStop` und in Amazon EC2 und beendet die Instance weiterhin. EC2

Sie können beispielsweise die EC2 Amazon-Konsole verwenden, um den Kündigungsschutz für eine EC2 Amazon-Instance in einem EMR-Cluster mit deaktiviertem Kündigungsschutz zu aktivieren. Wenn Sie den Cluster mit der Amazon EMR-Konsole, der oder der Amazon EMR-API beenden oder herunterskalieren AWS CLI, überschreibt Amazon EMR die `DisableApiTermination` Einstellung, setzt sie auf „False“ und beendet die Instance zusammen mit anderen Instances.

Sie können die EC2 Amazon-Konsole auch verwenden, um den Stop-Schutz für eine EC2 Amazon-Instance in einem EMR-Cluster mit deaktiviertem Kündigungsschutz zu aktivieren. Wenn Sie den Cluster beenden oder herunterskalieren, wird Amazon EMR in Amazon DisableApiStop auf False gesetzt EC2 und die Instance zusammen mit anderen Instances beendet.

Amazon EMR überschreibt die DisableApiStop Einstellung nur, wenn Sie einen Cluster beenden oder herunterskalieren. Wenn Sie den Kündigungsschutz in einem EMR-Cluster aktivieren oder deaktivieren, ändert Amazon EMR die disableApiStop Einstellung für keine der EC2 Instances im jeweiligen EMR-Cluster.

Important

Wenn Sie eine Instance als Teil eines Amazon EMR-Clusters mit Kündigungsschutz erstellen und die EC2 Amazon-API oder AWS CLI Befehle verwenden, um die Instance so zu ändern, dass das DisableApiTermination heißtfalse, und dann die EC2 Amazon-API oder AWS CLI Befehle den TerminateInstances Vorgang ausführen, wird die EC2 Amazon-Instance beendet.

Beendigungsschutz und instabile YARN-Knoten

Amazon EMR überprüft regelmäßig den Apache Hadoop YARN-Status von Knoten, die auf Kern- und EC2 Task-Amazon-Instances in einem Cluster ausgeführt werden. Der Gesundheitsstatus wird vom [NodeManager Health Checker](#) Service gemeldet. Wenn ein Knoten meldetUNHEALTHY, fügt der Amazon EMR-Instance-Controller den Knoten einer Denylist hinzu und weist ihm keine YARN-Container zu, bis er wieder fehlerfrei ist. Je nach Status des Kündigungsschutzes, des Austauschs fehlerhafter Knoten und der Amazon EMR-Release-Version [ersetzt Amazon EMR entweder die fehlerhafte Instance oder beendet die Zuweisung von Controllern zur Instance](#).

Kündigungsschutz und Kündigung nach Ausführung des Schritts

Wenn Sie die Kündigung nach der Ausführung des Schritts aktivieren und gleichzeitig den Kündigungsschutz aktivieren, ignoriert Amazon EMR den Kündigungsschutz.

Wenn Sie Schritte an einen Cluster übermitteln, können Sie die Eigenschaft ActionOnFailure festlegen, um zu bestimmen, was passiert, wenn die Ausführung eines Schritts aufgrund eines Fehlers nicht abgeschlossen werden kann. Die möglichen Werte für diese Einstellung sind TERMINATE_CLUSTER (TERMINATE_JOB_FLOW mit früheren Versionen), CANCEL_AND_WAIT und CONTINUE. Weitere Informationen finden Sie unter [Arbeit an einen Amazon EMR-Cluster einreichen](#).

Wenn ein Schritt fehlschlägt, der mit der `ActionOnFailure` Einstellung auf konfiguriert ist `CANCEL_AND_WAIT`, und wenn die Beendigung nach der Ausführung des Schritts aktiviert ist, wird der Cluster beendet, ohne dass nachfolgende Schritte ausgeführt werden.

Wenn ein Schritt fehlschlägt, für den `ActionOnFailure` auf `TERMINATE_CLUSTER` festgelegt wurde, können Sie anhand der folgenden Tabelle mit Einstellungen das Ergebnis ermitteln.

ActionOnFailure	Beendigung nach der Ausführung des Schritts	Termination protection	Ergebnis
TERMINATE_CLUSTER	Enabled	Disabled	Cluster wird beendet
	Aktiviert	Aktiviert	Cluster wird beendet
	Disabled	Aktiviert	Cluster wird weiter ausgeführt
	Disabled	Disabled	Cluster wird beendet

Beendigungsschutz und Spot Instances

Der Amazon EMR-Kündigungsschutz verhindert nicht, dass eine Amazon EC2 Spot-Instance beendet wird, wenn der Spot-Preis über den maximalen Spot-Preis steigt.

Konfigurieren des Beendigungsschutzes beim Starten eines Clusters

Sie können den Kündigungsschutz aktivieren oder deaktivieren, wenn Sie einen Cluster über die Konsole AWS CLI, die oder die API starten.

Für Cluster mit einem Knoten lauten die Standardeinstellungen für den Kündigungsschutz wie folgt:

- Starten eines Clusters über die Amazon EMR-Konsole — Termination Protection ist standardmäßig deaktiviert.
- Das Starten eines Clusters mit AWS CLI `aws emr create-cluster` — Termination Protection ist deaktiviert, sofern nicht anders angegeben `--termination-protected`.

- Starten eines Clusters mit dem Amazon [RunJobFlow](#) EMR-API-Befehl — Termination Protection ist deaktiviert, sofern der `TerminationProtected` boolesche Wert nicht auf `gesetzt ist. true`

Für Hochverfügbarkeitscluster lauten die Standardeinstellungen für den Kündigungsschutz wie folgt:

- Starten eines Clusters über die Amazon EMR-Konsole — Termination Protection ist standardmäßig aktiviert.
- Das Starten eines Clusters mit AWS CLI `aws emr create-cluster` — Termination Protection ist deaktiviert, sofern nicht anders angegeben `--termination-protected`.
- Starten eines Clusters mit dem Amazon [RunJobFlow](#) EMR-API-Befehl — Termination Protection ist deaktiviert, sofern der `TerminationProtected` boolesche Wert nicht auf `gesetzt ist. true`

Console

Um den Kündigungsschutz ein- oder auszuschalten, wenn Sie einen Cluster mit der Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie für EMR-Release-Version die Option `emr-6.6.0` oder höher aus.
4. Vergewissern Sie sich, dass unter Clusterbeendigung und Knotenersatz die Option Kündigungsschutz verwenden vorausgewählt ist, oder löschen Sie die Auswahl, um sie auszuschalten.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um den Kündigungsschutz ein- oder auszuschalten, wenn Sie einen Cluster mit dem AWS CLI

- Mit dem AWS CLI können Sie einen Cluster mit aktiviertem Kündigungsschutz mit dem `create-cluster` Befehl mit dem `--termination-protected` Parameter starten. Der Beendigungsschutz ist standardmäßig deaktiviert.

Im folgenden Beispiel wird ein Cluster mit aktiviertem Beendigungsschutz erstellt:

 Note

Linux-Zeilenumbruchzeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws emr create-cluster --name "TerminationProtectedCluster" --release-label emr-7.9.0 \
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --termination-protected
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Konfigurieren des Beendigungsschutzes für aktive Cluster

Sie können den Beendigungsschutz für einen aktiven Cluster mithilfe der Konsole oder AWS CLI konfigurieren.

Konsole

So schalten Sie den Kündigungsschutz für einen laufenden Cluster mit der Konsole ein oder aus

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Suchen Sie auf der Registerkarte Eigenschaften der Cluster-Detailseite nach Clusterbeendigung und wählen Sie Bearbeiten aus.
4. Aktivieren oder deaktivieren Sie das Kontrollkästchen Beendigungsschutz verwenden, um das Feature ein- oder auszuschalten. Wählen Sie dann zur Bestätigung Änderungen speichern aus.

AWS CLI

Um den Kündigungsschutz für einen laufenden Cluster ein- oder auszuschalten, verwenden Sie AWS CLI

- Um den Beendigungsschutz für einen ausgeführten Cluster über die AWS CLI zu aktivieren, verwenden Sie den Befehl `modify-cluster-attributes` mit dem Parameter `--termination-protected`. Um ihn zu deaktivieren, verwenden Sie den Parameter `--no-termination-protected`.

Das folgende Beispiel aktiviert den Kündigungsschutz auf dem Cluster mit der ID `j-3KVTXXXXXX7UG`:

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --termination-protected
```

Im folgenden Beispiel wird der Beendigungsschutz für dasselbe Cluster deaktiviert:

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
```

Ersetzen fehlerhafter Knoten durch Amazon EMR

Amazon EMR verwendet regelmäßig den [NodeManager Health Checker-Service](#) in Apache Hadoop, um den Status der Kernknoten in Ihrem Amazon EMR auf Amazon-Clustern zu überwachen. EC2 Wenn ein Knoten nicht optimal funktioniert, wird der Knoten als fehlerhaft markiert und der Health Checker meldet diesen Knoten an den Amazon EMR-Controller. Der Amazon EMR-Controller fügt den Knoten einer Ablehnungsliste hinzu und verhindert so, dass der Knoten neue YARN-Anwendungen empfängt, bis sich der Status des Knotens verbessert.

Note

Ein häufiger Grund dafür, dass ein Knoten fehlerhaft ist, ist, dass ihm nicht genügend Speicherplatz zur Verfügung steht. Für weitere Informationen darüber, wann ein Core-Knoten fast keinen Speicherplatz mehr hat, ist der folgende re:POST Knowledge Center-Artikel hilfreich: [Warum geht dem Core-Knoten in meinem Amazon EMR-Cluster der Speicherplatz aus?](#)

 Note

Hadoop bietet die Möglichkeit, benutzerdefinierte Node-Integritätsprüfungen durchzuführen. Dies wird in der Apache Hadoop-Dokumentation unter ausführlicher erklärt. [NodeManager](#)

Sie können wählen, ob Amazon EMR fehlerhafte Knoten beenden oder sie im Cluster belassen soll. Wenn Sie den Austausch fehlerhafter Knoten deaktivieren, bleiben diese auf der Sperrliste und werden weiterhin auf die Clusterkapazität angerechnet. Sie können zur Konfiguration und Wiederherstellung weiterhin eine Verbindung zu Ihrer Amazon EC2 Core-Instance herstellen, sodass Sie die Größe Ihres Clusters ändern können, wenn Sie Kapazität hinzufügen möchten. Weitere Informationen darüber, wie der Austausch und die Kündigung von Knoten funktionieren, finden Sie unter [Terminierungsschutz verwenden](#).

Wenn der Austausch fehlerhafter Knoten aktiviert ist, beendet Amazon EMR einen fehlerhaften Core-Knoten und stellt eine neue Instance bereit, basierend auf der Anzahl der Instances in der Instance-Gruppe oder basierend auf der Zielkapazität für Instance-Flotten. Wenn Knoten länger als 45 Minuten fehlerhaft sind, ersetzt Amazon EMR die [Knoten ordnungsgemäß](#). Wenn die ordnungsgemäße Außerbetriebnahme eines Knotens nicht innerhalb einer Stunde abgeschlossen ist, wird der Knoten gewaltsam beendet, es sei denn, durch seine Beendigung wird der Cluster unter den Replikationsfaktor oder die HDFS-Kapazitätsbeschränkungen gebracht.

 Important

Beachten Sie, dass sich die Zeit, die benötigt wird, bis ein Knoten ordnungsgemäß außer Betrieb genommen oder beendet wird, ändern kann.

Durch den Austausch fehlerhafter Knoten wird zwar die Wahrscheinlichkeit eines Datenverlusts erheblich verringert, das Risiko wird jedoch nicht vollständig ausgeschlossen. HDFS-Daten können beim ordnungsgemäßen Austausch einer fehlerhaften Core-Instance dauerhaft verloren gehen. Wir empfehlen, dass Sie Ihre Daten immer sichern.

Weitere Informationen zur Identifizierung fehlerhafter Knoten und zur Wiederherstellung finden Sie unter [Ressourcenfehler](#). Weitere bewährte Methoden, die Sie befolgen können, um die Integrität eines Clusters aufrechtzuerhalten, finden Sie in der folgenden Dokumentation zum Ressourcenfehler [Amazon EMR-Cluster terminates with NO_SLAVE_LEFT und Core Nodes FAILED_BY_MASTER](#).

Amazon EMR veröffentlicht Amazon CloudWatch Events für den Austausch fehlerhafter Knoten, sodass Sie verfolgen können, was mit Ihren fehlerhaften Core-Instances passiert. Weitere Informationen finden Sie unter [Ereignisse beim Austausch fehlerhafter Knoten](#).

Standardeinstellungen für den Austausch von Knoten und den Kündigungsschutz

Unhealthy Node Replacement ist für alle Amazon EMR-Versionen verfügbar, aber die Standardeinstellungen hängen von der von Ihnen gewählten Release-Bezeichnung ab. Sie können jede dieser Einstellungen ändern, indem Sie beim Erstellen eines neuen Clusters den Austausch fehlerhafter Knoten konfigurieren oder indem Sie jederzeit zur Cluster-Konfiguration wechseln.

Wenn Sie einen Einzelknoten-Cluster oder einen Hochverfügbarkeitscluster erstellen, auf dem Amazon EMR Version 7.0 oder niedriger ausgeführt wird, hängt die Standardeinstellung für den Austausch fehlerhafter Knoten vom Kündigungsschutz ab:

- Durch die Aktivierung des Kündigungsschutzes wird der Austausch fehlerhafter Knoten deaktiviert.
- Durch die Deaktivierung des Terminierungsschutzes wird der Austausch fehlerhafter Knoten ermöglicht.

Konfiguration des Austauschs fehlerhafter Knoten beim Start eines Clusters

Sie können den Austausch fehlerhafter Knoten aktivieren oder deaktivieren, wenn Sie einen Cluster über die Konsole AWS CLI, die oder die API starten.

Die Standardeinstellung für den Austausch fehlerhafter Knoten hängt davon ab, wie Sie den Cluster starten:

- Amazon EMR-Konsole — Der Austausch fehlerhafter Knoten ist standardmäßig aktiviert.
- AWS CLI `aws emr create-cluster` — Der Austausch fehlerhafter Knoten ist standardmäßig aktiviert, sofern Sie nichts anderes angeben. `--no-unhealthy-node-replacement`
- Der Amazon [RunJobFlow EMR-API-Befehl](#) — Austausch ungesunder Knoten ist standardmäßig aktiviert, sofern Sie den `UnhealthyNodeReplacement` booleschen Wert nicht auf oder setzen. `True False`

Console

Um den Austausch fehlerhafter Knoten ein oder aus zu schalten, wenn Sie mit der Konsole einen Cluster erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie für die EMR-Release-Version das gewünschte Amazon EMR-Release-Label aus.
4. Vergewissern Sie sich, dass unter Clusterbeendigung und Austausch von Knoten die Option Fehlerhafter Knotenaustausch (empfohlen) vorausgewählt ist, oder löschen Sie die Auswahl, um ihn auszuschalten.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um den Austausch fehlerhafter Knoten zu aktivieren oder zu deaktivieren, wenn Sie einen Cluster mit dem AWS CLI

- Mit dem können Sie einen Cluster starten AWS CLI, bei dem der Austausch fehlerhafter Knoten mit dem `create-cluster` Befehl mit dem `--unhealthy-node-replacement` Parameter aktiviert ist. Der Austausch fehlerhafter Knoten ist standardmäßig aktiviert.

Im folgenden Beispiel wird ein Cluster erstellt, bei dem der Austausch fehlerhafter Knoten aktiviert ist:

Note

Linux-Zeilenfortsetzungszeichen (`\`) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (`^`).

```
aws emr create-cluster --name "SampleCluster" --release-label emr-7.9.0 \  
--applications Name=Hadoop Name=Hive Name=Pig \
```

```
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \  
--instance-count 3 --unhealthy-node-replacement
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie unter [Amazon AWS CLI EMR-Befehle](#). AWS CLI

Konfiguration eines fehlerhaften Knotenaustauschs in einem laufenden Cluster

Sie können den Austausch fehlerhafter Knoten für einen laufenden Cluster mithilfe der Konsole, der oder der API ein AWS CLI- oder ausschalten.

Console

Um den Austausch fehlerhafter Knoten für einen laufenden Cluster mit der Konsole ein- oder auszuschalten

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Suchen Sie auf der Cluster-Detailseite auf der Registerkarte Eigenschaften nach Clusterbeendigung und Knotenersatz und wählen Sie Bearbeiten aus.
4. Aktivieren oder deaktivieren Sie das Kontrollkästchen für fehlerhaften Knotenaustausch, um die Funktion ein- oder auszuschalten. Wählen Sie dann zur Bestätigung Änderungen speichern aus.

AWS CLI

Um den Austausch fehlerhafter Knoten für einen laufenden Cluster ein- oder auszuschalten, verwenden Sie AWS CLI

- Um den Austausch fehlerhafter Knoten in einem laufenden Cluster mit dem zu aktivieren AWS CLI, verwenden Sie den `modify-cluster-attributes` Befehl mit dem `--unhealthy-node-replacement` Parameter. Um ihn zu deaktivieren, verwenden Sie den Parameter `--no-unhealthy-node-replacement`.

Im folgenden Beispiel wird der Austausch fehlerhafter Knoten auf dem Cluster mit der ID aktiviert: *j-3KVTXXXXXX7UG*

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --unhealthy-node-replacement
```

Im folgenden Beispiel wird der Austausch fehlerhafter Knoten auf demselben Cluster deaktiviert:

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-unhealthy-node-replacement
```

Arbeiten mit Amazon Linux AMIs in Amazon EMR

Amazon Linux Amazon-Maschinenabbilder (AMIs)

Amazon EMR verwendet ein Amazon Linux Amazon Machine Image (AMI), um EC2 Amazon-Instances zu initialisieren, wenn Sie einen Cluster erstellen und starten. Das AMI enthält das Amazon-Linux-Betriebssystem, weitere Software und die für die einzelnen Instances zum Hosten Ihrer Cluster-Anwendungen erforderlichen Konfigurationen.

Wenn Sie einen Cluster erstellen, verwendet Amazon EMR standardmäßig ein Amazon Linux (AL) -AMI, das speziell für die von Ihnen verwendete Amazon EMR-Release-Version erstellt wurde. Weitere Informationen zum standardmäßigen Amazon Linux AMI finden Sie unter [Verwenden des Standard-Amazon-Linux-AMI für Amazon EMR](#). Wenn Sie Amazon EMR 5.7.0 oder höher verwenden, können Sie für Amazon EMR ein benutzerdefiniertes Amazon Linux AMI anstelle des standardmäßigen Amazon Linux AMI angeben. Ein benutzerdefiniertes AMI ermöglicht Ihnen das Verschlüsseln des Root-Gerät-Datenträgers und das Anpassen von Anwendungen und Konfigurationen als Alternative zu Bootstrap-Aktionen. Sie können ein benutzerdefiniertes AMI für jeden Instance-Typ in der Instance-Gruppen- oder Instance-Flottenkonfiguration eines Amazon-EMR-Clusters angeben. Die Unterstützung mehrerer benutzerdefinierter AMIs gibt Ihnen die Flexibilität, mehr als einen Architekturtyp in einem Cluster zu verwenden. Siehe [Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration](#).

Amazon EMR fügt automatisch ein Amazon EBS-Allzweck-SSD-Volume als Root-Gerät für alle an. AMIs AMIs EBS-gestützte Leistungssteigerung. Weitere Informationen zu Amazon Linux AMIs finden

Sie unter [Amazon Machine Images \(AMI\)](#). Weitere Informationen zum Instance-Speicher für Amazon-EMR-Instances finden Sie unter [Speicheroptionen und Verhalten von Instances in Amazon EMR](#).

Themen

- [Verwenden des Standard-Amazon-Linux-AMI für Amazon EMR](#)
- [Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration](#)
- [Ändern Sie die Amazon Linux-Version, wenn Sie einen EMR-Cluster erstellen](#)
- [Anpassen des Root-Geräte-Volumes in Amazon EBS](#)

Verwenden des Standard-Amazon-Linux-AMI für Amazon EMR

Jede Amazon-EMR-Version verwendet ein standardmäßiges Amazon Linux AMI für Amazon EMR, es sei denn, Sie geben ein benutzerdefiniertes AMI an. Ab den Versionen Amazon EMR 5.36, Amazon EMR 6.6 und Amazon EMR 7.0 besteht das Standardverhalten für die Aktualisierung von Amazon Linux 2 (für EMR 5.x und 6.x, AL2 023 AL2 für EMR 7.x) in einem Amazon EMR-Standard-AMI darin, automatisch die neueste Amazon Linux-Version für das standardmäßige Amazon EMR-AMI anzuwenden.

Automatische Amazon-Linux-Updates für Amazon-EMR-Versionen

Wenn Sie einen Cluster mit der neuesten Patch-Version von Amazon EMR 7.0 oder höher, 6.6 oder höher oder 5.36 oder höher starten, verwendet Amazon EMR die neueste Amazon-Linux-Version für das standardmäßige Amazon-EMR-AMI. Zum Beispiel:

- Wenn es eine Version `x.x.0` und `x.x.1` gibt, erhält die `x.x.0`-Version beim `x.x.1`-Start keine AMI-Updates mehr.
- Ebenso erhält `x.x.1` beim Start von `x.x.2` keine AMI-Updates mehr.
- Später, wenn `x.y.0` veröffentlicht wird, erhält `x.x.[latest]` weiterhin AMI-Updates mit `x.y.[latest]`.

Um zu sehen, ob Sie die neueste Patch-Version verwenden, die durch die Zahl nach dem zweiten Dezimalpunkt (`6.8.1`) für eine Amazon-EMR-Version gekennzeichnet ist, sehen Sie sich die verfügbaren Versionen im [Amazon-EMR-Versionshandbuch](#) an, überprüfen Sie das Drop-down-Menü für Amazon-EMR-Versionen, wenn Sie einen Cluster in der Konsole erstellen, oder verwenden Sie die [ListReleaseLabels](#)-API- oder [list-release-labels](#)-CLI-Aktion. Um auf dem Laufenden

zu bleiben, wenn wir eine neue Amazon-EMR-Version veröffentlichen, abonnieren Sie den RSS-Feed auf der Seite [Was ist neu?](#) im Versionshandbuch.

Wenn Sie möchten, können Sie Ihren Cluster mit der Amazon-Linux-Version starten, mit der die Amazon-EMR-Version zuerst ausgeliefert wurde. Weitere Informationen zum Spezifizieren der Amazon-Linux-Version für Ihren Cluster finden Sie unter [Ändern Sie die Amazon Linux-Version, wenn Sie einen EMR-Cluster erstellen](#).

Standard-Amazon-Linux-Versionen

Themen

- [Standard AMIs für Amazon EMR 7.0 und höher](#)
- [Standard AMIs für Amazon EMR 6.6 und höher](#)
- [Standard AMIs für Amazon EMR 5.x](#)

Standard AMIs für Amazon EMR 7.0 und höher

In der folgenden Tabelle sind Amazon Linux-Informationen für die neueste Patch-Version der Amazon EMR-Versionen 7.0 und höher aufgeführt.

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.7.2 250527.1	6.1.134-152,225.amzn2023	19. Juni 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-central-1 • eu-west-1 • eu-west-2 • eu-south-1 • eu-west-3

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• eu-south-2 • eu-north-1 • eu-central-2 • ap-east-1 • ap-south-2 • ap-southeast-3 • ap-southeast-5 • ap-southeast-4 • ap-south-1 • ap-northeast-3 • ap-northeast-2 • ap-southeast-1 • ap-southeast-2 • ap-southeast-7 (7.1.0+) • ap-northeast-1 • af-south-1 • sa-east-1 • me-south-1 • me-central-1 • ca-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northwest-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• il-central-1 • mx-central-1 (7.1.0+)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.7.250512,0	6.1.134-152,225.amzn2023	04. Juni 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-central-1 • eu-west-1 • eu-west-2 • eu-south-1 • eu-west-3 • eu-south-2 • eu-north-1 • eu-central-2 • ap-east-1 • ap-south-2 • ap-southeast-3 • ap-southeast-5 • ap-southeast-4 • ap-south-1 • ap-northeast-3 • ap-northeast-2 • ap-southeast-1 • ap-southeast-2 • ap-southeast-7 (7.1.0+) • ap-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• af-south-1 • sa-east-1 • me-south-1 • me-central-1 • ca-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northwest-1 • il-central-1 • mx-central-1 (7.1.0+)

Verwenden des Standard-AMI 375

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• af-south-1 • sa-east-1 • me-south-1 • me-central-1 • ca-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northwest-1 • il-central-1 • mx-central-1 (7.1.0+)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.7.250414,0	6.1.132-147.221.amzn2023	12. Mai 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • ap-southeast-7 • ap-southeast-5 • mx-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.7.250331,0	6.1.131-143.221.amzn2023	18. April 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • ap-southeast-7 • ap-southeast-5 • mx-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.6.250303.0	6.1.129-138.220.amzn2023	27. März 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • ap-southeast-7 • ap-southeast-5 • mx-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.6.2 250218.2	6.1.128-136.201.amzn2023	08. März 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • ap-southeast-7 • ap-southeast-5 • mx-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.6.250211.0	6.1.127-135.201.amzn2023	26. Februar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • ap-southeast-7 • ap-southeast-5 • mx-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.6.250123.4	6.1.124-134.200.amzn2023	27. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.6.250115.0	6.1.119-129.201.amzn2023	23. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.6.241121,0	6.1.115-126,197.amzn2023	12. Dezember 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.5.241031,0	6.1.112-124.190.amzn2023	15. November 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.5.241001,zn2023	6.1.109-118.189.amzn2023	4. Oktober 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.5.240819.0	6.1.102-111.182.amzn2023	20. August 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.5.2023,5.240730,(	6.1.97-104.177.amzn2023	2. August 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.5.240722.0	6.1.97-104.177.amzn2023	24. Juli 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.5.240708.0	6.1.96-102.177.amzn2023	23. Juli 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.3.240304,0	6.1.79-99.164.amzn2023	12. März 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.3.240219.0	6.1.77-99.164.amzn2023	1. März 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.3.240205,0	6.1.75-99.163.amzn2023	19. Februar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.3.240122.0	6.1.72-96.amzn2023	5. Februar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.3.240108,0	6.1.72-96.amzn2023	24. Januar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2023.3.2 231211.4	6.1.66-91.160.amzn2023	19. Dezember 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

Standard AMIs für Amazon EMR 6.6 und höher

In der folgenden Tabelle sind Amazon-Linux-Informationen für die neueste Patch-Version von Amazon EMR, 6.6.x und höher, aufgeführt.

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 527.1	4.14.355-277.647.amzn2	19. Juni 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-central-1 • eu-west-1 • eu-west-2 • eu-south-1 • eu-west-3

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• eu-south-2 (6.7.0+) • eu-north-1 • eu-central-2 (6,7,0+) • ap-east-1 • ap-south-2 (6,7,0+) • ap-southeast-3 • ap-southeast-5 (6.11.1 und 6.14.0+) • ap-southeast-4 (6.8.0+) • ap-south-1 • ap-northeast-3 • ap-northeast-2 • ap-southeast-1 • ap-southeast-2 • ap-southeast-7 (6.11.1 und 6.14.0+) • ap-northeast-1 • af-south-1 • sa-east-1 • me-south-1 • me-central-1 (6.7.0+) • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• ca-west-1 (6,9,0+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northwest-1 • il-central-1 (6,7,0+) • mx-central-1 (6.11.1 und 6.14.0+)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025512.0	4.14.355-277.643.amzn2	04. Juni 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-central-1 • eu-west-1 • eu-west-2 • eu-south-1 • eu-west-3 • eu-south-2 (6,7,0+) • eu-north-1 • eu-central-2 (6,7,0+) • ap-east-1 • ap-south-2 (6,7,0+) • ap-southeast-3 • ap-southeast-5 (6.11.1 und 6.14.0+) • ap-southeast-4 (6.8.0+) • ap-south-1 • ap-northeast-3 • ap-northeast-2 • ap-southeast-1 • ap-southeast-2

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• ap-southeast-7 (6.11.1 und 6.14.0+) • ap-northeast-1 • af-south-1 • sa-east-1 • me-south-1 • me-central-1 (6.7.0+) • ca-central-1 • ca-west-1 (6,9,0+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northwest-1 • il-central-1 (6,7,0+) • mx-central-1 (6.11.1 und 6.14.0+)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 428.0	4.14.355-276.639.amzn2	23. Mai 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-central-1 • eu-west-1 • eu-west-2 • eu-south-1 • eu-west-3 • eu-south-2 (6.7.0+) • eu-north-1 • eu-central-1-2 (6,7,0+) • ap-east-1 • ap-south-2 (6,7,0+) • ap-southeast-3 • ap-southeast-5 (6.11.1 und 6.14.0+) • ap-southeast-4 (6.8.0+) • ap-south-1 • ap-northeast-3 • ap-northeast-2 • ap-southeast-1 • ap-southeast-2

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• ap-southeast-7 (6.11.1 und 6.14.0+) • ap-northeast-1 • af-south-1 • sa-east-1 • me-south-1 • me-central-1 (6.7.0+) • ca-central-1 • ca-west-1 (6,9,0+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northwest-1 • il-central-1 (6,7,0+) • mx-central-1 (6.11.1 und 6.14.0+)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025414.0	4.14.355-276.618.amzn2	12. Mai 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6.1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 321.0	4,14.355	09. April 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025305.0	4,14,355	18. März 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 220.0	4,14.355	08. März 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 201.0	4,14.355	28. Februar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 123.4	4,14.355	27. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025116.0	4,14.355	23. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 217.0	4,14.355	8. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024001.0	4,14.352	4. Oktober 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10.1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• sa-east-1 • me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 816.0	4,14.350	21. August 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 809.0	4,14,349	20. August 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10.1+) • eu-south-1 • eu-south-2 (6.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6.1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• sa-east-1 • me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 719.0	4,14.348	25. Juli 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024709.1	4,14.348	23. Juli 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 223.0	4,14.336	8. März 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 131.0	4,14.336	14. Februar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10.1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• sa-east-1 • me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 124.0	4,14.336	7. Februar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 109.0	4,14.334	24. Januar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10.1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• sa-east-1 • me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 218.0	4,14.330	2. Januar 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 206.0	4,14.330	22. Dezember 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 116.0	4,14.328	11. Dezember 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 101.0	4,14.327	17. November 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023.020.1	4,14.326	07. November 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023012.1	4,14.326	26. Oktober 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 926.0	4,14.322	19. Oktober 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.8+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.20238906,0	4,14.322	04. Oktober 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.9+ und 5.36.1)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 822.0	4,14.322	30. August 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.9+ und 5.36.1)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 808,0	4,14.320	24. August 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.9+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 727.0	4,14.320	14. August 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.9+ und 5.36.1)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 719,0	4,14.320	02. August 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-southeast-4 (6.8+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10+) • me-south-1 • ca-central-1 • il-central-1 (6.9+ und 5.36.1)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 628.0	4,14.318	12. Juli 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6.10+)

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			- me-south-1 - ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 612,0	4,14.314	23. Juni 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6.10+)

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			- me-south-1 - ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 504.1	4,14.313	16. Mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10+) • eu-south-1 • eu-south-2 (6.10+) • ap-east-1 • ap-south-1 • ap-south-2 (6.10+) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 418,0	4,14.311	3. Mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (nur 6.10) • eu-south-1 • eu-south-2 (nur 6.10) • ap-east-1 • ap-south-1 • ap-south-2 (nur 6.10) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			- me-central-1 - me-south-1 - ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 404.1	4,14.311	18. April 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• <code>ca-central-1</code>
2.0.2023 404.0	4,14.311	10. April 2023	• <code>us-east-1</code> • <code>eu-west-3</code>

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 320.0	4,14.309	30. März 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 307,0	4,14.305	15. März 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 207,0	4,14.304	03. März 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• ca-central-1
2.0.2023 119.1	4,14.301	9. Februar 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 210.1	4.14.301	12. Januar 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 103.3	4,14.296	5. Dezember 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022004,0	4,14.294	02. November 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 912.1	4,14.291	7. Oktober 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1
2.0.2022 805.0	4,14.287	30. August 2022	• us-west-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 719.0	4,14.287	10. August 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 426,0	4,14.281	10. Juni 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 406.1	4,14.275	2. Mai 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

Standard AMIs für Amazon EMR 5.x

In der folgenden Tabelle sind Amazon-Linux-Informationen für die neueste Patch-Version von Amazon EMR 5.x, Version 5.36 und höher, aufgeführt.

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 428.0	4.14.355-276.639.amzn2	23. Mai 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-central-1 • eu-west-1 • eu-west-2 • eu-south-1 • eu-west-3 • eu-south-2 • eu-north-1 • eu-central-2 • ap-east-1 • ap-south-2 • ap-southeast-3 • ap-southeast-5 (5.36,2) • ap-southeast-4 • ap-south-1 • ap-northeast-3 • ap-northeast-2 • ap-southeast-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• ap-southeast-2 • ap-southeast-7 (5,36,2) • ap-northeast-1 • af-south-1 • sa-east-1 • me-south-1 • me-central-1 • ca-central-1 • ca-west-1 (5,36,1+) • us-gov-east-1 (5,36,1+) • us-gov-west-1 (5,36,1+) • cn-north-1 (5,36,1+) • cn-northeast-1 (5,36,1+) • il-central-1 • mx-central-1 (5,36,2)

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025321.0	4,14.355	09. April 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 305.0	4,14,355	18. März 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 220.0	4,14.355	08. März 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 201.0	4,14.355	28. Februar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 123.4	4,14.355	27. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2025 116.0	4,14.355	23. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 217.0	4,14.355	8. Januar 2025	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 (6.10.1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-south-1 • ca-central-1 • il-central-1 • ca-west-1 (6.9.0+ und 5.36.1+) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 816.0	4,14.350	21. August 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024809.0	4,14,349	20. August 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6.10.1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• sa-east-1 • me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024 719.0	4,14.348	25. Juli 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2024709.1	4,14.348	23. Juli 2024	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,1+) • eu-south-1 • eu-south-2 (6,1+) • ap-east-1 • ap-south-1 • ap-south-2 (6,1+) • ap-southeast-3 • ap-southeast-4 (6.8.1+ und 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsReleaseLabel (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
			• me-central-1 (6.10.1+) • me-south-1 • ca-central-1 • il-central-1 (6.8.1+ und 5.36.1) • ca-west-1 (6.9.1+ und 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 504.1	4,14.313	16. Mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 418,0	4,14.311	3. Mai 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • me-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 404.1	4,14.311	18. April 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1
2.0.2023 404.0	4,14.311	10. April 2023	• us-east-1 • eu-west-3

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 320.0	4,14.309	30. März 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 307,0	4,14.305	15. März 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2023 207,0	4,14.304	03. März 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 210.1	4.14.301	12. Januar 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 103.3	4,14.296	5. Dezember 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022004,0	4,14.294	02. November 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 912.1	4,14.291	7. Oktober 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 719.0	4,14.287	10. August 2022	• us-west-1 • eu-west-3 • eu-north-1 • eu-central-1 • ap-south-1 • me-south-1

OsRelease Label (AL-Version)	AL-Kernel-Version	Verfügbarkeitsdatum	AWS-Regionen
2.0.2022 426,0	4,14.281	14. Juni 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

Überlegungen zu Softwareupdates

Beachten Sie die folgenden Standardverhalten für Softwareupdates:

Amazon EMR 7.x – Amazon Linux 2023

Amazon EMR-Versionen 7.0 und höher laufen auf Amazon Linux 2023 (AL2023). Ihre Cluster enthalten nur Sicherheitsupdates, die in der Version von AL2 023 AMI verfügbar waren, die Sie bei der Erstellung ausgewählt haben. Beim Start installieren EMR-Cluster-Instances nicht die neuesten Sicherheitsupdates aus den aktivierten Paket-Repositorys. Um die neuesten Sicherheitsupdates zu erhalten, empfehlen wir Ihnen, Ihren Cluster regelmäßig neu zu erstellen. Weitere Informationen zu AL2 023 finden Sie unter [Aktualisieren von Amazon Linux 2023](#) im Amazon Linux 2023-Benutzerhandbuch.

Amazon EMR 5.36+ und 6.6+ — Amazon Linux 2

Die Amazon EMR-Versionen 5.36 und höher sowie 6.6 und höher laufen auf Amazon Linux 2 (AL2). Ihre Cluster enthalten nur Sicherheitsupdates, die in der AL2 AMI-Version verfügbar waren, die Sie bei der Erstellung ausgewählt haben. Beim Start installieren EMR-Cluster-Instances nicht die neuesten Sicherheitsupdates aus den aktivierten Paket-Repositorys. Um die neuesten Sicherheitsupdates zu erhalten, empfehlen wir Ihnen, Ihren Cluster regelmäßig neu zu erstellen. Weitere Informationen dazu finden Sie unter [Software auf Ihrer AL2 Instance verwalten](#) im Amazon Linux 2-Benutzerhandbuch. AL2

Amazon EMR 3.x, 4.x, 5.0.0 bis 5.35.0 und 6.0.0 bis 6.5.0 — Amazon Linux AMI ist an die Amazon EMR-Release-Version gebunden

Für die Amazon EMR-Versionen 3.x, 4.x, 5.0.0 bis 5.35.0 und 6.0.0 bis 6.5.0 basiert das Standard-AMI auf dem meisten up-to-date Amazon Linux-AMI, das zum Zeitpunkt der Amazon EMR-Version verfügbar war. Das AMI wurde auf Kompatibilität mit den Big-Data-Anwendungen und Amazon EMR-Funktionen getestet, die in dieser Release-Version enthalten sind.

Wenn eine EC2 Amazon-Instance zum ersten Mal in einem Cluster gestartet wird, der auf dem standardmäßigen Amazon Linux (AL) oder AL2 AMI für Amazon EMR basiert, sucht sie in den aktivierten Paket-Repositorys für AL und Amazon EMR nach Softwareupdates, die für die Release-Version gelten. Wie bei anderen EC2 Amazon-Instances, auf denen AL oder ausgeführt wird AL2 AMIs, werden kritische und wichtige Sicherheitsupdates aus diesen Repositorys automatisch installiert.

Beachten Sie auch, dass Ihre Netzwerkkonfiguration den HTTP- und HTTPS-Ausgang zu Amazon-Linux-Repositorys in Amazon S3 zulassen muss. Andernfalls schlagen Sicherheitsupdates fehl. Weitere Informationen finden Sie unter [Amazon Linux — Package Repository](#) im EC2 Amazon-Benutzerhandbuch. Standardmäßig sind andere Softwarepakete und Kernel-Updates, die einen

Neustart erfordern, einschließlich NVIDIA und CUDA, vom automatischen Download beim ersten Start ausgeschlossen.

 Important

Die Amazon EMR-Versionen 3.x, 4.x, 5.0.0 bis 5.35.0 und 6.0.0 bis 6.5.0 verwenden das Standardverhalten von Amazon Linux und laden wichtige und kritische Kernel-Updates, die einen Neustart erfordern, nicht automatisch herunter und installieren sie. Dies ist dasselbe Verhalten wie bei anderen EC2 Amazon-Instances, auf denen das Standard-AL oder ausgeführt wird AL2 AMIs. Wenn neue Amazon-Linux-Softwareupdates, die einen Neustart erfordern (wie Kernel-, NVIDIA- und CUDA-Updates), nach der Veröffentlichung einer Amazon-EMR-Version verfügbar werden, laden EMR-Cluster-Instances, die das Standard-AMI ausführen, diese Updates nicht automatisch herunter und installieren sie. Um Kernel-Updates zu erhalten, können Sie das [neueste Amazon Linux AMI](#) verwenden, wie unter [Verwenden eines benutzerdefinierten AMI beschrieben, um mehr Flexibilität für die Amazon EMR-Clusterkonfiguration zu bieten](#).

Der Cluster wird mit oder ohne Updates gestartet

Beachten Sie, dass die Cluster-Instance ihren Start trotzdem abschließt, wenn Softwareupdates nicht installiert werden können, weil Paket-Repositorys beim ersten Clusterstart nicht erreichbar sind. Beispielsweise sind Repositorys möglicherweise nicht erreichbar, weil S3 vorübergehend nicht verfügbar ist, oder Sie haben möglicherweise VPC- oder Firewallregeln so konfiguriert, dass sie den Zugriff blockieren.

sudo yum update nicht ausführen

Wenn Sie eine Verbindung mit einer Cluster-Instance über SSH herstellen, enthalten die ersten Zeilen der Bildschirmausgabe einen Link zu den Versionshinweisen für das von der Instance verwendete Amazon-Linux-AMI, einen Hinweis auf die jeweils aktuelle Amazon-Linux-AMI-Version, einen Hinweis auf die Anzahl der für das Update verfügbaren Pakete in den aktivierten Repositorys und eine Anweisung für die Ausführung von `sudo yum update`.

 Important

Es wird nachdrücklich davon abgeraten, `sudo yum update` auf Cluster-Instances auszuführen, weder über eine SSH-Verbindung noch über eine Bootstrap-Aktion. Dies kann zu Inkompatibilitäten führen, da alle Pakete unterschiedslos installiert werden.

Bewährte Methoden für Softwareupdates

Bewährte Methoden für die Verwaltung von Software-Updates

- Wenn Sie eine ältere Version von Amazon EMR verwenden, sollten Sie eine Migration auf die neueste Version in Betracht ziehen und testen, bevor Sie Softwarepakete aktualisieren.
- Wenn Sie auf eine höhere Version migrieren oder Softwarepakete upgraden, sollten Sie die Implementierung zunächst in einer Umgebung außerhalb der Produktion testen. Die Option zum Klonen von Clustern über die Amazon-EMR-Managementkonsole ist für diese Zwecke nützlich.
- Sie sollten die Software-Updates für Ihre Anwendungen und Ihre Version des Amazon-Linux-AMI einzeln bewerten. Testen und installieren Sie nur Pakete in Produktionsumgebungen, die Ihrer Meinung nach für Sicherheit, Anwendungsfunktionalität oder Leistung unbedingt notwendig sind.
- Achten Sie im [Amazon-Linux-Sicherheitszentrum](#) auf Updates.
- Vermeiden Sie die Installation von Paketen über SSH-Verbindungen mit einzelnen Cluster-Instances. Verwenden Sie stattdessen eine Bootstrap-Aktion, um Pakete auf allen Cluster-Instances wie notwendig zu installieren und zu aktualisieren. Hierzu müssen Sie einen Cluster beenden und neu starten. Weitere Informationen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#).

Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration

Wenn Sie Amazon EMR 5.7.0 oder höher verwenden, können Sie für Amazon EMR ein benutzerdefiniertes Amazon Linux AMI anstelle des standardmäßigen Amazon Linux AMI angeben. Ein benutzerdefiniertes AMI ist nützlich, wenn Sie Folgendes durchführen möchten:

- Installieren Sie Anwendungen vorab und führen Sie weitere Anpassungen aus, statt Bootstrap-Aktionen zu verwenden. Dies kann die Cluster-Startzeit verbessern und den Startup-

Workflow optimieren. Weitere Informationen sowie ein Beispiel finden Sie unter [Erstellen eines benutzerdefinierten Amazon-Linux-AMI aus einer vorkonfigurierten Instance](#).

- Implementierung komplexerer Cluster- und Knoten-Konfigurationen als von Bootstrap-Aktionen zugelassen.
- Verschlüsseln Sie die EBS-Root-Geräte-Volumes (Boot-Volumes) der EC2 Instances in Ihrem Cluster, wenn Sie eine Amazon EMR-Version unter 5.24.0 verwenden. Wie beim Standard-AMI beträgt die Mindestgröße des Root-Volumes für ein benutzerdefiniertes AMI 10 GiB für die Amazon-EMR-Versionen 6.9 und niedriger sowie 15 GiB für die Amazon-EMR-Versionen 6.10 und höher. Weitere Informationen finden Sie unter [Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Amazon-EBS-Root-Gerät-Datenträger](#).

 Note

Ab Amazon EMR Version 5.24.0 können Sie eine Sicherheitskonfigurationsoption verwenden, um EBS-Root-Geräte und Speichervolumes zu verschlüsseln, wenn Sie dies als Ihren Schlüsselanbieter angeben AWS KMS . Weitere Informationen finden Sie unter [Verschlüsselung lokaler Datenträger](#).

Ein benutzerdefiniertes AMI muss in derselben AWS Region vorhanden sein, in der Sie den Cluster erstellen. Es sollte auch der EC2 Instance-Architektur entsprechen. Eine m5.xlarge-Instance hat beispielsweise eine x86_64-Architektur. Um ein m5.xlarge mithilfe eines benutzerdefinierten AMI bereitzustellen, sollte Ihr benutzerdefiniertes AMI daher auch über eine x86_64-Architektur verfügen. Ebenso sollte Ihr benutzerdefiniertes AMI eine arm64-Architektur haben, um eine m6g.xlarge-Instance bereitzustellen, die über eine arm64-Architektur verfügt. Weitere Informationen zur Identifizierung eines Linux-AMI für Ihren Instance-Typ [finden Sie unter Suchen Sie ein Linux-AMI](#) im EC2 Amazon-Benutzerhandbuch.

 Important

EMR-Cluster, auf denen Amazon Linux oder Amazon Linux 2 Amazon Machine Images (AMIs) ausgeführt werden, verwenden das Standardverhalten von Amazon Linux und laden wichtige und kritische Kernel-Updates, die einen Neustart erfordern, nicht automatisch herunter und installieren sie. Dies ist dasselbe Verhalten wie bei anderen EC2 Amazon-Instances, auf denen das standardmäßige Amazon Linux-AMI ausgeführt wird. Wenn neue Amazon-Linux-Softwareupdates, die einen Neustart erfordern (wie Kernel-, NVIDIA- und CUDA-Updates), nach der Veröffentlichung einer Amazon-EMR-Version verfügbar

werden, laden EMR-Cluster-Instances, die das Standard-AMI ausführen, diese Updates nicht automatisch herunter und installieren sie. Um Kernel-Updates zu erhalten, können Sie [Ihr Amazon-EMR-AMI](#) so anpassen, dass es [das neueste Amazon-Linux-AMI verwendet](#).

Erstellen eines benutzerdefinierten Amazon-Linux-AMI aus einer vorkonfigurierten Instance

Die grundlegenden Schritte für das Vorinstallieren von Software und das Ausführen weiterer Konfigurationen zur Erstellung eines benutzerdefinierten Amazon-Linux-AMI für Amazon EMR sind:

- Starten Sie eine Instance über das Amazon-Linux-Basis-AMI.
- Stellen Sie eine Verbindung mit der Instance her, um Software zu installieren und andere Anpassungen vorzunehmen.
- Erstellen Sie ein neues Abbild (AMI-Snapshot) der Instance, die Sie konfiguriert haben.

Nachdem Sie das Abbild auf der Grundlage Ihrer benutzerdefinierten Instance erstellt haben, können Sie es auf ein verschlüsseltes Ziel kopieren, wie im Abschnitt [Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Amazon-EBS-Root-Gerät-Datenträger](#) beschrieben.

Tutorial: Erstellen eines AMI aus einer Instance mit installierter eigener Software

Um eine EC2 Instance zu starten, die auf dem neuesten Amazon Linux AMI basiert

1. Verwenden Sie den AWS CLI , um den folgenden Befehl auszuführen, der eine Instance aus einem vorhandenen AMI erstellt. *MyKeyName* Ersetzen Sie durch das key pair, das Sie für die Verbindung *MyAmiId* mit der Instance verwenden, und durch die ID eines entsprechenden Amazon Linux AMI. Das neueste AMI finden Sie IDs unter [Amazon Linux AMI](#).

Note

Linux-Zeilenumbruchzeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws ec2 run-instances --image-id MyAmiID \  
--count 1 --instance-type m5.xlarge \  
--key-name MyKeyName --region us-west-2
```

Der Ausgabewert `InstanceId` wird im nächsten Schritt als *MyInstanceId* verwendet.

2. Führen Sie den folgenden Befehl aus:

```
aws ec2 describe-instances --instance-ids MyInstanceId
```

Der Ausgabewert `PublicDnsName` wird im nächsten Schritt verwendet, um eine Verbindung mit der Instance herzustellen.

So stellen Sie eine Verbindung mit der Instance her und installieren Software

1. Verwenden Sie eine SSH-Verbindung, mit der Sie Shell-Befehle auf Ihrer Linux-Instance ausführen können. Weitere Informationen finden Sie unter [Herstellen einer Verbindung zu Ihrer Linux-Instance mithilfe von SSH](#) im EC2 Amazon-Benutzerhandbuch.
2. Führen Sie alle erforderlichen Anpassungen durch. Zum Beispiel:

```
sudo yum install MySoftwarePackage  
sudo pip install MySoftwarePackage
```

So erstellen Sie einen Snapshot vom benutzerdefinierten Abbild

- Nachdem Sie die Instance angepasst haben, erstellen Sie mithilfe des Befehls `create-image` ein AMI von der Instance.

```
aws ec2 create-image --no-dry-run --instance-id MyInstanceId --name MyEmrCustomAmi
```

Der Ausgabewert `imageID` wird verwendet, wenn Sie den Cluster starten oder einen verschlüsselten Snapshot erstellen. Weitere Informationen erhalten Sie unter [Ein einzelnes benutzerdefiniertes AMI in einem EMR-Cluster verwenden](#) und [Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Amazon-EBS-Root-Gerät-Datenträger](#).

So verwenden Sie ein benutzerdefiniertes AMI in einem Amazon-EMR-Cluster

Sie können ein benutzerdefiniertes AMI verwenden, um einen Amazon-EMR-Cluster auf zwei Arten bereitzustellen:

- Verwenden Sie ein einziges benutzerdefiniertes AMI für alle EC2 Instances im Cluster.
- Verwenden Sie unterschiedliche benutzerdefinierte AMIs Einstellungen für die verschiedenen EC2 Instance-Typen, die im Cluster verwendet werden.

Sie können bei der Bereitstellung eines EMR-Clusters nur eine der beiden Optionen verwenden, und Sie können sie nicht mehr ändern, nachdem der Cluster gestartet wurde.

Überlegungen zur Verwendung eines einzelnen oder mehrerer benutzerdefinierter Elemente AMIs in einem Amazon EMR-Cluster

Überlegungen	Einfaches benutzerd efiniertes AMI	Mehrfach benutzerdefiniert AMIs
Verwenden Sie sowohl x86- als auch Graviton2-Prozessoren mit Custom AMIs im selben Cluster	✗ Nicht unterstützt	✓ Wird unterstützt
Die AMI-Anpassung variiert je nach Instance-Typ	✗ Nicht unterstützt	✓ Wird unterstützt
Ändern Sie die benutzerdefinierte Version AMIs , wenn Sie eine neue Task-Instanz hinzufügen. groups/fleets to a running cluster. Note: you cannot change the custom AMI of existing instance groups/fleets	✗ Nicht unterstützt	✓ Wird unterstützt
Verwenden Sie die AWS Konsole, um einen Cluster zu starten	✓ Wird unterstützt	✗ Nicht unterstützt
Wird verwendet AWS CloudForm ation , um einen Cluster zu starten	✓ Wird unterstützt	✓ Wird unterstützt

Ein einzelnes benutzerdefiniertes AMI in einem EMR-Cluster verwenden

Verwenden Sie eine der folgenden Optionen, um eine benutzerdefinierte AMI-ID anzugeben, wenn Sie einen Cluster erstellen:

- AWS Management Console
- AWS CLI
- Amazon-EMR-SDK
- Amazon EMR-API [RunJobFlow](#)
- AWS CloudFormation (siehe die CustomAmiID Eigenschaft unter [Cluster InstanceGroupConfig](#), [Cluster InstanceTypeConfig](#) InstanceGroupConfig, [Ressource](#) oder [Ressource InstanceFleetConfig - InstanceTypeConfig](#))

Amazon EMR console

So geben Sie ein einzelnes benutzerdefiniertes AMI über die Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Suchen Sie unter Name und Anwendungen nach Betriebssystemoptionen. Wählen Sie Benutzerdefiniertes AMI und geben Sie Ihre AMI-ID in das Feld Benutzerdefiniertes AMI ein.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um ein einzelnes benutzerdefiniertes AMI anzugeben mit dem AWS CLI

- Verwenden Sie den Parameter `--custom-ami-id` zum Angeben der AMI-ID, wenn Sie den Befehl `aws emr create-cluster` ausführen.

Im folgenden Beispiel wird ein Cluster angegeben, der ein einzelnes benutzerdefiniertes AMI mit einem 20 GiB-Boot-Volume verwendet. Weitere Informationen finden Sie unter [Anpassen des Root-Geräte-Volumes in Amazon EBS](#).

 Note

Linux-Zeilenfortsetzungszeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws emr create-cluster --name "Cluster with My Custom AMI" \  
--custom-ami-id MyAmiID --efs-root-volume-size 20 \  
--release-label emr-5.7.0 --use-default-roles \  
--instance-count 2 --instance-type m5.xlarge
```

Verwenden Sie mehrere benutzerdefinierte AMIs in einem Amazon EMR-Cluster

Um einen Cluster mit mehreren benutzerdefinierten Clustern zu erstellen AMIs, verwenden Sie eine der folgenden Optionen:

- AWS CLI Version 1.20.21 oder höher
- AWS SDK
- Amazon EMR [RunJobFlow](#) in der Amazon EMR-API-Referenz
- AWS CloudFormation (siehe die CustomAmiID Eigenschaft unter [Cluster InstanceGroupConfig](#), [Cluster InstanceTypeConfig](#) InstanceGroupConfig, [Ressource](#) oder [Ressource InstanceFleetConfig](#) -) InstanceTypeConfig

Die AWS Management Console unterstützt derzeit nicht die Erstellung eines Clusters mit mehreren benutzerdefinierten Clustern AMIs.

Example - Verwenden Sie die AWS CLI, um einen Instanzgruppen-Cluster mit mehreren benutzerdefinierten zu erstellen AMIs

Mit der AWS CLI-Version 1.20.21 oder höher können Sie dem gesamten Cluster ein einzelnes benutzerdefiniertes AMI zuweisen, oder Sie können jedem Instance-Knoten in Ihrem Cluster mehrere benutzerdefinierte AMIs zuweisen.

Das folgende Beispiel zeigt einen einheitlichen Instance-Gruppen-Cluster, der mit zwei Instance-Typen (m5.xlarge) erstellt wurde, die für alle Knotentypen (Primär, Core, Aufgabe) verwendet werden. Jeder Knoten hat mehrere benutzerdefinierte Knoten. AMIs Das Beispiel veranschaulicht mehrere Features der Konfiguration mit mehreren benutzerdefinierten AMIs:

- Auf Cluster-Ebene ist kein benutzerdefiniertes AMI zugewiesen. Dadurch sollen Konflikte zwischen mehreren benutzerdefinierten AMIs und einem einzelnen benutzerdefinierten AMI vermieden werden, die dazu führen würden, dass der Clusterstart fehlschlägt.
- Der Cluster kann mehrere benutzerdefinierte Knoten für AMIs Primär-, Kern- und einzelne Taskknoten haben. Dies ermöglicht individuelle AMI-Anpassungen, wie z. B. vorinstallierte Anwendungen, ausgefeilte Cluster-Konfigurationen und verschlüsselte Amazon EBS-Root-Geräte-Volumes.
- Der Core-Knoten der Instance-Gruppe kann nur einen Instance-Typ und ein entsprechendes benutzerdefiniertes AMI haben. Ebenso kann der Primärknoten nur einen Instance-Typ und ein entsprechendes benutzerdefiniertes AMI haben.
- Der Cluster kann mehrere Aufgabenknoten haben.

```
aws emr create-cluster --instance-groups
InstanceGroupType=PRIMARY,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567
InstanceGroupType=TASK,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-456789
```

Example - Verwenden Sie die AWS CLI-Version 1.20.21 oder höher, um einem laufenden Instanzgruppen-Cluster mit mehreren Instanztypen und mehreren benutzerdefinierten Instanztypen einen Task-Knoten hinzuzufügen AMIs

Mit der AWS CLI-Version 1.20.21 oder höher können Sie einer Instanzgruppe, AMIs die Sie einem laufenden Cluster hinzufügen, mehrere benutzerdefinierte Instanzen hinzufügen. Das CustomAmiId-Argument kann zusammen mit dem add-instance-groups-Befehl verwendet werden, wie im folgenden Beispiel gezeigt. Beachten Sie, dass dieselbe mehrfache benutzerdefinierte AMI-ID (ami-123456) in mehr als einem Knoten verwendet wird.

```
aws emr create-cluster --instance-groups
InstanceGroupType=PRIMARY,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567
```

```
{
  "ClusterId": "j-123456",
  ...
}
```

```
aws emr add-instance-groups --cluster-id j-123456 --instance-groups
  InstanceGroupType=Task,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
```

Example - Verwenden Sie die AWS CLI-Version 1.20.21 oder höher, um einen Instance-Flottencluster, mehrere benutzerdefinierte AMIs, mehrere Instance-Typen, On-Demand-Primärinstanzen, On-Demand-Kern, mehrere Kern- und Taskknoten zu erstellen

```
aws emr create-cluster --instance-fleets
  InstanceFleetType=PRIMARY,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}']
  InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}',
    '{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
  InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-567890}',
    '{InstanceType=m6g.xlarge, CustomAmiId=ami-567890}']
```

Example - Verwenden Sie die AWS CLI-Version 1.20.21 oder höher, um einem laufenden Cluster mit mehreren Instanztypen und mehreren benutzerdefinierten Task-Knoten hinzuzufügen AMIs

```
aws emr create-cluster --instance-fleets
  InstanceFleetType=PRIMARY,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}']
  InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}',
    '{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']

{
  "ClusterId": "j-123456",
  ...
}
```

```
aws emr add-instance-fleet --cluster-id j-123456 --instance-fleet
  InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-567890}',
    '{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
```

Verwalten von Updates für AMI-Paket-Repositorys

Beim ersten Start stellt Amazon Linux standardmäßig eine AMIs Verbindung zu Paket-Repositorys her, um Sicherheitsupdates zu installieren, bevor andere Dienste gestartet werden. Je nach Ihren

Anforderungen können Sie diese Aktualisierungen deaktivieren, wenn Sie ein benutzerdefiniertes AMI für Amazon EMR angeben. Die Option zum Deaktivieren dieser Funktion steht nur verfügbar, wenn Sie ein benutzerdefiniertes AMI verwenden. Standardmäßig werden Amazon-Linux-Kernel-Updates und andere Softwarepakete, die einen Neustart erfordern, nicht aktualisiert. Beachten Sie, dass Ihre Netzwerkkonfiguration den HTTP- und HTTPS-Ausgang zu Amazon-Linux-Repositorys in Amazon S3 zulassen muss, da andernfalls Sicherheitsupdates nicht erfolgreich sein werden.

Warning

Wir empfehlen dringend, dass Sie beim Neustart alle installierten Pakete aktualisieren, wenn Sie ein benutzerdefiniertes AMI angeben. Wenn Sie keine Pakete aktualisieren, entstehen zusätzliche Sicherheitsrisiken.

Mit der AWS Management Console können Sie die Option zum Deaktivieren von Updates auswählen, wenn Sie Benutzerdefiniertes AMI wählen.

Mit dem können Sie angeben `AWS CLI --repo-upgrade-on-boot NONE, --custom-ami-id` wann Sie den `create-cluster` Befehl verwenden möchten.

Mit der Amazon EMR-API können Sie `NONE` den [RepoUpgradeOnBoot](#) Parameter angeben.

Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Amazon-EBS-Root-Gerät-Datenträger

Um den Amazon-EBS-Root-Gerät-Datenträger eines Amazon-Linux-AMI für Amazon EMR zu verschlüsseln, kopieren Sie ein Snapshot-Abbild von einem unverschlüsselten AMI zu einem verschlüsselten Ziel. Informationen zum Erstellen verschlüsselter EBS-Volumes finden Sie unter [Amazon EBS-Verschlüsselung](#) im EC2 Amazon-Benutzerhandbuch. Als Quell-AMI für den Snapshot können Sie das Amazon-Linux-Basis-AMI verwenden. Sie können auch einen Snapshot von einem AMI kopieren, das vom angepassten Amazon-Linux-AMI abgeleitet wurde.

Note

Ab Amazon EMR Version 5.24.0 können Sie eine Sicherheitskonfigurationsoption verwenden, um EBS-Root-Geräte und Speichervolumes zu verschlüsseln, wenn Sie dies als Ihren Schlüsselanbieter angeben `AWS KMS`. Weitere Informationen finden Sie unter [Verschlüsselung lokaler Datenträger](#).

Sie können einen externen Schlüsselanbieter oder einen AWS KMS-Schlüssel verwenden, um das EBS-Root-Volume zu verschlüsseln. Die von Amazon EMR verwendete Servicerolle (normalerweise die Standard-EMR_DefaultRole) muss mindestens zum Verschlüsseln und Entschlüsseln des Volumes berechtigt sein, damit Amazon EMR einen Cluster mit dem AMI erstellen kann. Bei der Verwendung AWS KMS als Schlüsselanbieter bedeutet dies, dass die folgenden Aktionen zulässig sein müssen:

- kms:encrypt
- kms:decrypt
- kms:ReEncrypt*
- kms>CreateGrant
- kms:GenerateDataKeyWithoutPlaintext"
- kms:DescribeKey"

Hierfür fügen Sie am einfachsten die Rolle als Schlüsselbenutzer wie im folgenden Tutorial beschrieben hinzu. Die folgende Richtlinienanweisung dient als Beispiel für den Fall, dass Sie Rollenrichtlinien anpassen müssen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EmrDiskEncryptionPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms>CreateGrant",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Tutorial: Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Root-Gerät-Datenträger mithilfe eines KMS-Schlüssels

Im ersten Schritt dieses Beispiels muss der ARN eines KMS-Schlüssels ermittelt oder neu erstellt werden. Weitere Informationen zum Erstellen von -Schlüsseln finden Sie unter [Erstellen von Schlüsseln](#) im AWS Key Management Service -Entwicklerhandbuch. Im folgenden Verfahren wird gezeigt, wie Sie als Schlüsselbenutzer die Standard-Servicerolle `EMR_DefaultRole` zur Schlüsselrichtlinie hinzufügen. Notieren Sie sich den ARN-Wert für den Schlüssel, wenn Sie ihn erstellen oder bearbeiten. Sie verwenden den ARN später, wenn Sie das AMI erstellen.

Um die Servicerolle für Amazon EC2 zur Liste der Benutzer von Verschlüsselungsschlüsseln mit der Konsole hinzuzufügen

1. Melden Sie sich bei der Konsole AWS Key Management Service (AWS KMS) an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/kms>.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie den Alias des zu verwendenden KMS-Schlüssel aus.
4. Wählen Sie auf der Seite mit den Schlüsseldetails unter Key Users (Schlüsselbenutzer) die Option Add (Hinzufügen) aus.
5. Wählen Sie im Dialogfeld Anfügen die Amazon-EMR-Servicerolle aus. Der Name der Standardrolle lautet `EMR_DefaultRole`.
6. Wählen Sie Anfügen aus.

Um ein verschlüsseltes AMI zu erstellen mit dem AWS CLI

- Verwenden Sie den `aws ec2 copy-image` Befehl von AWS CLI , um ein AMI mit einem verschlüsselten EBS-Root-Geräte-Volumen und dem Schlüssel, den Sie geändert haben, zu erstellen. Ersetzen Sie den angegebenen Wert `--kms-key-id` durch den vollständigen ARN des Schlüssels, den Sie zuvor erstellt oder geändert haben.

Note

Linux-Zeilenumbruchzeichen (`\`) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (`^`).

```
aws ec2 copy-image --source-image-id MyAmiId \  
--source-region us-west-2 --name MyEncryptedEMRAmi \  
--encrypted --kms-key-id arn:aws:kms:us-west-2:12345678910:key/xxxxxxxx-xxxx-xxxx-  
xxxx-xxxxxxxxxxxxxxxx
```

Die Ausgabe des Befehls enthält die ID des AMI, das Sie erstellt haben, welches Sie angeben können, wenn Sie einen Cluster erstellen. Weitere Informationen finden Sie unter [Ein einzelnes benutzerdefiniertes AMI in einem EMR-Cluster verwenden](#). Sie können dieses AMI auch anpassen, indem Sie Software installieren und andere Konfigurationen durchführen. Weitere Informationen finden Sie unter [Erstellen eines benutzerdefinierten Amazon-Linux-AMI aus einer vorkonfigurierten Instance](#).

Bewährte Methoden und Überlegungen

Wenn Sie ein benutzerdefiniertes AMI für Amazon EMR erstellen, sollten Sie Folgendes bedenken:

- Die Amazon EMR 7.x-Serie basiert auf Amazon Linux 2023. Für diese Amazon EMR-Versionen müssen Sie benutzerdefinierte AMIs Images verwenden, die auf Amazon Linux 2023 basieren. Informationen zum Suchen eines benutzerdefinierten Basis-AMI finden Sie unter [Suchen eines Linux-AMI](#).
- Für Amazon EMR-Versionen unter 7.x wird Amazon Linux 2023 nicht AMIs unterstützt.
- Amazon EMR 5.30.0 und höher sowie die Amazon-EMR-6.x-Serie basieren auf Amazon Linux 2. Für diese Amazon EMR-Versionen müssen Sie benutzerdefinierte AMIs Images verwenden, die auf Amazon Linux 2 basieren. Informationen zum Suchen eines benutzerdefinierten Basis-AMI finden Sie unter [Suchen eines Linux-AMI](#).
- Für Amazon EMR-Versionen unter 5.30.0 und 6.x wird Amazon Linux 2 AMIs nicht unterstützt.
- Sie müssen ein 64-Bit-Amazon-Linux-AMI verwenden. Ein 32-Bit-AMI wird nicht unterstützt.
- Amazon Linux AMIs mit mehreren Amazon EBS-Volumes wird nicht unterstützt.
- Legen Sie Ihrer Anpassung das neueste von EBS gestützte [Amazon Linux AMI](#) zugrunde. Eine Liste von Amazon Linux AMIs und dem entsprechenden AMI IDs finden Sie unter [Amazon Linux AMI](#).
- Kopieren Sie keinen Snapshot einer vorhandenen Amazon-EMR-Instance, um ein benutzerdefiniertes AMI zu erstellen. Das verursacht Fehler.

- Es werden nur die mit Amazon EMR kompatiblen HVM-Virtualisierungstypen und Instances unterstützt. Stellen Sie sicher, dass Sie während der AMI-Anpassung ein HVM-Abbild und einen Instance-Typ auswählen, das bzw. der mit Amazon EMR kompatibel ist. Kompatible Instances und Virtualisierungstypen finden Sie unter [Unterstützte Instance-Typen mit Amazon EMR](#).
- Die Servicerolle muss über Startberechtigungen für das AMI verfügen. Das AMI muss also entweder öffentlich sein, oder Sie müssen der Eigentümer des AMI sein, oder das AMI wurde vom Eigentümer für Sie freigegeben.
- Wenn Sie Benutzer im AMI erstellen, deren Namen mit Anwendungsnamen übereinstimmen (z. B. hadoop, hdfs, yarn oder spark), führt das zu Fehlern.
- Der Inhalt von /tmp, /var und /emr – sofern im AMI vorhanden – wird während des Startup entsprechend nach /mnt/tmp, /mnt/var und /mnt/emr verschoben. Dateien werden beibehalten; bei großen Mengen an Daten kann jedoch der Startup länger als erwartet dauern.
- Wenn Sie ein benutzerdefiniertes Amazon-Linux-AMI verwenden, das auf einem Amazon-Linux-AMI mit einem Erstellungsdatum vom 8.11.2018 basiert, kann der Oozie-Server nicht gestartet werden. Wenn Sie Oozie verwenden, erstellen Sie ein benutzerdefiniertes AMI, das auf einer Amazon-Linux-AMI-ID mit einem anderen Erstellungsdatum basiert. Sie können den folgenden AWS CLI Befehl verwenden, um eine Liste mit Images IDs für alle HVM Amazon Linux AMIs mit einer Version 2018.03 zusammen mit dem Veröffentlichungsdatum zurückzugeben, sodass Sie ein geeignetes Amazon Linux-AMI als Basis auswählen können. MyRegion Ersetzen Sie es durch Ihre Regionskennung, z. B. us-west-2.

```
aws ec2 --region MyRegion describe-images --owner amazon --query 'Images[?
Name!=`null`][?starts_with(Name, `amzn-ami-hvm-2018.03`) == `true`].
[CreationDate,ImageId,Name]' --output text | sort -rk1
```

- In Fällen, in denen Sie eine VPC mit einem nicht standardmäßigen Domainnamen und AmazonProvided DNS verwenden, sollten Sie die rotate Option in der DNS-Konfiguration des Betriebssystems nicht verwenden.
- Wenn Sie ein benutzerdefiniertes AMI erstellen, das den Amazon EC2 Systems Manager (SSM) -Agenten enthält, kann der aktivierte SSM-Agent einen Bereitstellungsfehler auf dem Cluster verursachen. Um dies zu vermeiden, deaktivieren Sie den SSM-Agenten, wenn Sie ein benutzerdefiniertes AMI verwenden. Um dies zu tun, deaktivieren Sie bei der Auswahl und dem Start Ihrer EC2 Amazon-Instance den SSM-Agenten, bevor Sie die Instance verwenden, um ein benutzerdefiniertes AMI zu erstellen, und anschließend Ihren EMR-Cluster erstellen.

Weitere Informationen finden Sie unter [Erstellen eines Amazon EBS-gestützten Linux-AMI](#) im EC2 Amazon-Benutzerhandbuch.

Ändern Sie die Amazon Linux-Version, wenn Sie einen EMR-Cluster erstellen

Wenn Sie einen Cluster mit Amazon EMR 6.6.0 oder höher starten, verwendet er automatisch die neueste Amazon-Linux-2-Version, die für das standardmäßige Amazon-EMR-AMI validiert wurde. Sie können eine andere Amazon Linux-Version für Ihren Cluster mit der Amazon-EMR-Konsole oder dem AWS CLI angeben.

Amazon EMR console

So ändern Sie die Amazon-Linux-Version, wenn Sie einen Cluster über die Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie für EMR-Version die Option emr-6.6.0 oder höher aus.
4. Wählen Sie unter Betriebssystemoptionen die Option Amazon Linux-Version aus, deaktivieren Sie das Kontrollkästchen Automatisch die neuesten Amazon Linux-Updates anwenden und wählen Sie die gewünschte Amazon Linux-Version aus.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um die Amazon-Linux-Version zu ändern, wenn Sie einen Cluster mit AWS CLI erstellen

- Verwenden Sie den `--os-release-label`-Parameter, um die Amazon-Linux-Version anzugeben, wenn Sie den Befehl `aws emr create-cluster` ausführen.

```
aws emr create-cluster --name "Cluster with Different Amazon Linux Release" \
--os-release-label 2.0.20210312.1 \
--release-label emr-6.6.0 --use-default-roles \
--instance-count 2 --instance-type m5.xlarge
```

Anpassen des Root-Geräte-Volumes in Amazon EBS

Sie können Ihren Volumetyp und andere Attribute je nach Anwendungsfall und Kostenanforderungen festlegen. Sie können Standardwerte akzeptieren oder Anpassungen vornehmen.

EBS-Root-Volume-StandardEinstellungen

Mit Amazon EMR 4.x und höher können Sie die Größe des Root-Volumes angeben, wenn Sie einen Cluster erstellen. Mit Amazon-EMR-Versionen 6.15.0 und höher können Sie auch die IOPS und den Durchsatz des Root-Volumes angeben. Die Attribute gelten nur für das Root-Geräte-Volume in Amazon EBS und gelten für alle Instances im Cluster. Die Attribute gelten nicht für Speicher-Volumes, die Sie separat für jeden Instance-Typ beim Erstellen Ihres Clusters angeben.

- Die Standardgröße des Root-Volumes wurde in Amazon EMR 6.10.0 und höher auf 15 GB erhöht. Frühere Versionen haben eine Standardgröße für das Root-Volume von 10 GiB. Sie können dies auf bis zu 100 GiB einstellen.
- Der Standardwert für Root-Volume-IOPS ist 3 000. Sie können dies auf bis zu 16 000 einstellen.
- Der Standarddurchsatz für das Root-Volume beträgt 125MiB/s. You can adjust this up to 1000 Mib/s.

Note

Das Verhältnis zwischen Größe und IOPS des Root-Volumes darf nicht höher als 1 Volume zu 500 IOPS (1:500) sein, während das Verhältnis zwischen IOPS und Durchsatz des Root-Volumes nicht höher als 1 IOPS zu 0,25 Durchsatz (1:0,25) sein darf.

Weitere Informationen zu Amazon EBS finden Sie unter [Amazon EC2 Root Device Volume](#).

Root-Gerät-Volumetyp mit dem Standard-AMI

Wenn Sie das Standard-AMI verwenden, wird der Volumetyp des Root-Geräts durch die von Ihnen verwendete Amazon-EMR-Version bestimmt.

- Mit Amazon-EMR-Versionen 6.15.0 und höher fügt Amazon EMR Allzweck-SSD (gp3) als Volumetyp für das Root-Gerät an.
- Mit Amazon-EMR-Versionen unter 6.15.0 fügt Amazon EMR Allzweck-SSD (gp2) als Volumetyp für das Root-Gerät an.

Root-Gerät-Volumetyp mit dem benutzerdefinierten AMI

Ein benutzerdefiniertes AMI kann andere Root-Gerät-Datenträgertypen haben. Amazon EMR verwendet immer Ihren benutzerdefinierten AMI-Volumetyp.

- Mit den Amazon-EMR-Versionen 6.15.0 und höher können Sie die Größe des Root-Volumes, die IOPS und den Durchsatz für Ihr benutzerdefiniertes AMI konfigurieren, sofern diese Attribute für den benutzerdefinierten AMI-Volumetyp gelten.
- Bei Amazon-EMR-Versionen unter 6.15.0 können Sie nur die Größe des Root-Volumes für Ihr benutzerdefiniertes AMI konfigurieren.

Wenn Sie bei der Erstellung Ihres Clusters keine Root-Volume-Größe, IOPS oder Durchsatz konfigurieren, verwendet Amazon EMR gegebenenfalls die Werte aus dem benutzerdefinierten AMI. Wenn Sie sich entscheiden, diese Werte bei der Erstellung Ihres Clusters zu konfigurieren, verwendet Amazon EMR die von Ihnen angegebenen Werte, sofern die Werte mit dem benutzerdefinierten AMI-Root-Volume kompatibel sind und von diesem unterstützt werden. Weitere Informationen finden Sie unter [Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration](#).

Preise nach Root-Gerät-Volumegröße

Die Kosten des EBS-Root-Gerät-Datenträgers wird anteilig nach Stunde berechnet. Dies erfolgt auf der Grundlage der monatlichen EBS-Gebühren für den Volume-Typ in der Region, in der der Cluster ausgeführt wird. Gleiches gilt für Speicher-Volumes. Die Gebühren gelten für GB, aber da Sie die Größe des Root-Volumes in GiB angeben, sollten Sie dies bei Ihren Kostenschätzungen berücksichtigen (1 GB entspricht 0,931323 GiB).

Allzweck-SSDs gp2 und gp3 werden unterschiedlich abgerechnet. Verwenden Sie die folgenden Formeln, um die Kosten für EBS-Root-Gerät-Datenträger in Ihrem Cluster zu schätzen:

Allzweck-SSD gp2

Die Kosten für gp2 beinhalten nur die Größe des EBS-Volumes in GB.

```
($EBS size in GB/month) * 0.931323 / 30 / 24 * EMR_EBSRootVolumesizeInGiB *  
InstanceCount
```

Nehmen Sie als Beispiel einen Cluster mit einem Primärknoten, einem Core-Knoten und einem Amazon Linux AMI mit einer Standardgröße des Root-Gerät-Volumes von 10 GiB. Wenn die EBS-

Kosten in der Region 0,10 GB/month, that works out to be approximately \$0.00129 per instance per hour, and \$0.00258 per hour for the cluster (\$0.10/GB/month USD/geteilt durch 30 Tage, geteilt durch 24 Stunden, multipliziert mit 10 GB, multipliziert mit 2 Cluster-Instances betragen).

Allzweck-SSD gp3

Die Kosten für gp3 beinhalten die Größe des EBS-Volumes in GB, IOPS über 3000 (3000 IOPS kostenlos) und einen Durchsatz von über 125 (kostenlos). MB/s (125 MB/s

```
($EBS size in GB/month) * 0.931323 / 30 / 24 * EMR_EBSRootVolumesizeInGiB *
InstanceCount
+
($EBS IOPS/Month)/30/24* (EMR_EBSRootVolumeIops - 3000) * InstanceCount
+
($EBS throughput/Month)/30/24* (EMR_EBSRootVolumeThroughputInMb/s - 125) *
InstanceCount
```

Nehmen Sie als Beispiel einen Cluster mit einem Primärknoten, einem Core-Knoten und einem Amazon Linux AMI mit einer Standardgröße des Root-Gerät-Volumes von 15 GiB, 4 000 IOPS und 140 Durchsatz. Wenn die EBS-Kosten in der Region 0,10 USD/mehr als 125 USD betragen. GB/month, \$0.005/provisioned IOPS/month over 3000, and \$0.040/provisioned MB/s/month Das entspricht ungefähr 0,009293 USD pro Instance und Stunde und 0,018586 USD pro Stunde für den Cluster.

Festlegen benutzerdefinierter Root-Gerät-Volume-Einstellungen

Note

Das Verhältnis zwischen Größe und IOPS des Root-Volumes darf nicht höher als 1 Volume zu 500 IOPS (1:500) sein, während das Verhältnis zwischen IOPS und Durchsatz des Root-Volumes nicht höher als 1 IOPS zu 0,25 Durchsatz (1:0,25) sein darf.

Console

So geben Sie Volumeattribute für Amazon-EBS-Root-Geräte über die Amazon-EMR-Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.

2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie Amazon EMR Version 6.15.0 oder höher aus.
4. Navigieren Sie unter Cluster configuration (Cluster-Konfiguration) zum Abschnitt EBS root volume und geben Sie einen Wert für eines der Attribute ein, die Sie konfigurieren möchten.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

So geben Sie die Volumeattribute des Amazon-EBS-Root-Geräts mit der AWS CLI an

- Verwenden Sie die Parameter `--ebs-root-volume-size`, `--ebs-root-volume-iops` und `--ebs-root-volume-throughput` des Befehls [create-cluster](#) wie im folgenden Beispiel gezeigt.

Note

Linux-Zeilenfortsetzungszeichen (`\`) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (`^`).

```
aws emr create-cluster --release-label emr-6.15.0\  
--ebs-root-volume-size 20 \  
--ebs-root-volume-iops 3000\  
--ebs-root-volume-throughput 135\  
--instance-groups InstanceGroupType=MASTER,\  
InstanceCount=1,InstanceType=m5.xlarge  
InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge
```

Konfigurieren Sie Anwendungen, wenn Sie Ihren Amazon EMR-Cluster starten

Wenn Sie ein Software-Release auswählen, verwendet Amazon EMR ein Amazon Machine Image (AMI) mit Amazon Linux zur Installation der Software, die Sie beim Starten Ihres Clusters ausgewählt

haben (z. B. Hadoop, Spark und Hive). Amazon EMR bietet regelmäßig neue Versionen mit neuen Features, neuen Anwendungen und allgemeine Aktualisierungen. Wir empfehlen, dass Sie die neueste Version zum Starten Ihres Clusters verwenden (sofern möglich). Die neueste Version ist die Standardoption beim Starten eines Clusters über die Konsole.

Weitere Informationen zu Amazon-EMR-Versionen und den mit jeder Version verfügbaren Softwareversionen finden Sie im [Amazon-EMR-Versionshandbuch](#). Weitere Informationen zum Bearbeiten der Standardkonfigurationen von auf Ihrem Cluster installierten Anwendungen und Software finden Sie unter [Konfigurieren von Anwendungen](#) im Amazon-EMR-Versionshandbuch. Einige Versionen der Open-Source-Komponenten von Hadoop und Spark in Amazon-EMR-Versionen nutzen Patches und Verbesserungen, die im [Amazon-EMR-Versionshandbuch](#) dokumentiert sind.

Zusätzlich zur standardmäßigen Software und den zur Installation auf Ihrem Cluster verfügbaren Anwendungen können Sie mit Bootstrap-Aktionen benutzerdefinierte Software installieren. Bootstrap-Aktionen sind Skripts, die beim Start Ihres Clusters und beim Ausführen neuer, bei der Erstellung des Clusters hinzugefügter Knoten in den Instances ausgeführt werden. Bootstrap-Aktionen sind auch nützlich, um AWS CLI Befehle auf jedem Knoten aufzurufen, um Objekte von Amazon S3 auf jeden Knoten in Ihrem Cluster zu kopieren.

Note

Die Verwendung von Bootstrap-Aktionen unterscheidet sich in Amazon-EMR-Version 4.x und höher. Weitere Informationen zu diesen Unterschieden zu den Amazon-EMR-AMI-Versionen 2.x und 3.x finden Sie unter [In 4.x eingeführte Unterschiede](#) im Amazon-EMR-Versionshandbuch.

Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren

Sie können eine Bootstrap-Aktion verwenden, um zusätzliche Software zu installieren oder die Konfiguration von Cluster-Instances anzupassen. Bootstrap-Aktionen sind Skripts, die auf Clustern ausgeführt werden, nachdem Amazon EMR die Instance mithilfe von Amazon Machine Image (AMI) startet. Bootstrap-Aktionen werden ausgeführt, bevor Amazon EMR die Anwendungen installiert, die Sie bei der Erstellung des Clusters angeben haben und bevor die Cluster-Knoten mit der Bearbeitung der Daten beginnen. Wenn Sie einem aktiven Cluster Knoten hinzufügen, werden die Bootstrap-Aktionen auf diesen Knoten auch auf die gleiche Weise ausgeführt. Sie können benutzerdefinierte Bootstrap-Aktionen erstellen und sie beim Erstellen Ihres Clusters angeben.

Die meisten vordefinierten Bootstrap-Aktionen für die Amazon-EMR-AMI-Versionen 2.x und 3.x werden in den Versionen 4.x von Amazon EMR nicht unterstützt. Beispielsweise werden `configure-Hadoop` und `configure-daemons` in Amazon-EMR-Version 4.x nicht unterstützt. Stattdessen stellt Amazon-EMR-Version 4.x diese Funktionalität nativ bereit. Weitere Informationen zur Migration von Bootstrap-Aktionen von den Amazon-EMR-AMI-Versionen 2.x und 3.x auf Amazon-EMR-Version 4.x finden Sie unter [Anpassen der Cluster- und Anwendungskonfiguration mit früheren AMI-Versionen von Amazon EMR](#) im Amazon-EMR-Versionshandbuch.

Bootstrap-Aktionen – Grundlagen

Bootstrap-Aktionen werden standardmäßig als Hadoop-Benutzer ausgeführt. Sie können eine Bootstrap-Aktion mit Root-Berechtigungen ausführen, indem Sie `sudo` verwenden.

Alle Amazon-EMR-Verwaltungsschnittstellen unterstützen Bootstrap-Aktionen. Sie können bis zu 16 Bootstrap-Aktionen pro Cluster angeben, indem Sie mehrere `bootstrap-actions` Parameter von der Konsole oder API aus angeben. AWS CLI

Von der Amazon-EMR-Konsole können Sie optional eine Bootstrap-Aktion beim Erstellen eines Clusters angeben.

Wenn Sie die CLI verwenden, können Sie Verweise auf Bootstrap-Aktionsskripts an Amazon EMR übergeben, indem Sie beim Erstellen des Clusters den Parameter `--bootstrap-actions` mit dem Befehl `create-cluster` hinzufügen.

```
--bootstrap-actions Path="s3://amzn-s3-demo-bucket/filename",Args=[arg1,arg2]
```

Wenn die Bootstrap-Aktion einen Fehlercode ungleich null zurückgibt, wird dieser von Amazon EMR wie ein Fehler behandelt und die Instance wird beendet. Wenn zu viele Instances bei ihren Bootstrap-Aktionen fehlschlagen, beendet Amazon EMR den Cluster. Wenn nur wenige Instances ausfallen, versucht Amazon EMR, die ausgefallenen Instances neu zuzuordnen und fortzufahren. Verwenden Sie den Cluster-Fehlercode `lastStateChangeReason`, um Fehler zu identifizieren, die durch eine Bootstrap-Aktion verursacht wurden.

Eine bedingte eine Bootstrap-Aktion ausführen

Um Bootstrap-Aktionen nur auf dem Hauptknoten auszuführen, können Sie eine benutzerdefinierte Bootstrap-Aktion mit etwas Logik verwenden, um festzustellen, ob es sich bei dem Knoten um einen Hauptknoten handelt.

```
#!/bin/bash
if grep isMaster /mnt/var/lib/info/instance.json | grep false;
then
    echo "This is not master node, do nothing, exiting"
    exit 0
fi
echo "This is master, continuing to execute script"
# continue with code logic for master node below
```

Die folgende Ausgabe wird von einem Core-Knoten aus gedruckt.

```
This is not master node, do nothing, exiting
```

Die folgende Ausgabe wird vom Hauptknoten aus gedruckt.

```
This is master, continuing to execute script
```

Um diese Logik zu verwenden, laden Sie Ihre Bootstrap-Aktion, einschließlich des obigen Codes, in Ihren Amazon-S3-Bucket hoch. Fügen Sie auf der AWS CLI den `--bootstrap-actions` Parameter zum `aws emr create-cluster` API-Aufruf hinzu und geben Sie den Speicherort Ihres Bootstrap-Skripts als Wert von `an. Path`.

Aktionen beim Herunterfahren

Ein Bootstrap-Aktionsskript kann eine oder mehrere Shutdown-Aktionen durchführen, indem es Skripts in das Verzeichnis `/mnt/var/lib/instance-controller/public/shutdown-actions/` schreibt. Wenn ein Cluster beendet wird, werden alle Skripts in diesem Verzeichnis parallel ausgeführt. Jedes Skript muss innerhalb von 60 Sekunden ausgeführt und abgeschlossen werden.

Es wird nicht garantiert, dass Shutdown-Aktionsskripts ausgeführt werden, wenn der Knoten mit einem Fehler beendet wird.

Note

Bei Verwendung der Amazon-EMR-Version 4.0 und höher müssen Sie das Verzeichnis `/mnt/var/lib/instance-controller/public/shutdown-actions/` auf dem Hauptknoten manuell erstellen. Es ist standardmäßig zwar nicht vorhanden, nach Erstellung

werden die Skripts in diesem Verzeichnis aber trotzdem vor dem Herunterfahren ausgeführt. Weitere Informationen zum Herstellen einer Verbindung mit dem Master-Knoten zum Erstellen von Verzeichnissen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Benutzerdefinierte Bootstrap-Aktionen verwenden

Sie können ein benutzerdefiniertes Skript erstellen, um eine angepasste Bootstrap-Aktion auszuführen. Alle Amazon-EMR-Schnittstellen können auf eine benutzerdefinierte Bootstrap-Aktion verweisen.

Note

Für eine optimale Leistung empfehlen wir, benutzerdefinierte Bootstrap-Aktionen, -Skripts und andere Dateien, die Sie mit Amazon EMR verwenden möchten, in einem Amazon S3 S3-Bucket zu speichern, der sich in demselben Cluster AWS-Region befindet.

Inhalt

- [Benutzerdefinierte Bootstrap-Aktionen hinzufügen](#)
- [Verwenden einer benutzerdefinierten Bootstrap-Aktion zum Kopieren eines Objekts aus Amazon S3 in jeden Knoten](#)

Benutzerdefinierte Bootstrap-Aktionen hinzufügen

Console

Um mit der Konsole einen Cluster mit einer Bootstrap-Aktion zu erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Bootstrap-Aktionen die Option Hinzufügen aus, um einen Namen, einen Skriptspeicherort und optionale Argumente für Ihre Aktion anzugeben. Wählen Sie Bootstrap-Aktion hinzufügen aus.

4. Fügen Sie optional weitere Bootstrap-Aktionen hinzu.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um einen Cluster mit einer benutzerdefinierten Bootstrap-Aktion zu erstellen, verwenden Sie AWS CLI

Wenn Sie die Aktion AWS CLI zum Einbeziehen einer Bootstrap-Aktion verwenden, geben Sie Path und Args als kommagetrennte Liste an. Bei dem folgenden Beispiel wird keine Argumentliste verwendet.

- Um einen Cluster mit einer benutzerdefinierten Bootstrap-Aktion zu starten, geben Sie den folgenden Befehl ein und *myKey* ersetzen Sie ihn durch den Namen Ihres EC2 key pair. Fügen Sie `--bootstrap-actions` als Parameter ein und geben Sie den Speicherort Ihres Bootstrap-Skripts als Wert von Path an.
- Linux-, UNIX- und Mac OS X-Benutzer:

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://elasticmapreduce/bootstrap-actions/download.sh"
```

- Windows-Nutzer:

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --use-
default-roles --ec2-attributes KeyName=myKey --applications Name=Hive Name=Pig
--instance-count 3 --instance-type m5.xlarge --bootstrap-actions Path="s3://
elasticmapreduce/bootstrap-actions/download.sh"
```

Wenn Sie die Instance-Anzahl ohne den `--instance-groups`-Parameter angeben, wird ein einzelner Primärknoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

 Note

Wenn Sie noch nicht die standardmäßige Amazon EMR-Servicerolle und das EC2 Instanzprofil erstellt haben, geben Sie ein, um sie `aws emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl eingeben.

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Verwenden einer benutzerdefinierten Bootstrap-Aktion zum Kopieren eines Objekts aus Amazon S3 in jeden Knoten

Sie können mit einer Bootstrap-Aktion Objekte von Amazon S3 in jeden Knoten eines Cluster kopieren, bevor Ihre Anwendungen installiert werden. Der AWS CLI ist auf jedem Knoten eines Clusters installiert, sodass Ihre Bootstrap-Aktion Befehle aufrufen AWS CLI kann.

Das folgende Beispiel zeigt ein einfaches Skript für eine Bootstrap-Aktion, die die Datei `myfile.jar` aus Amazon S3 zum lokalen Ordner `/mnt1/myfolder` auf jedem Cluster-Knoten kopiert. Das Skript wird mit dem Dateinamen `copymyfile.sh` in Amazon S3 mit den folgenden Inhalten gespeichert.

```
#!/bin/bash
aws s3 cp s3://amzn-s3-demo-bucket/myfilefolder/myfile.jar /mnt1/myfolder
```

Wenn Sie den Cluster starten, geben Sie das Skript an. Das folgende AWS CLI Beispiel verdeutlicht dies:

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.9.0 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://amzn-s3-demo-bucket/myscriptfolder/copymyfile.sh"
```

Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren

Ein wichtiger Aspekt bei der Erstellung eines Amazon EMR-Clusters ist die Konfiguration von EC2 Amazon-Instances und Netzwerkoptionen. Dieses Kapitel behandelt diese Optionen im Detail und beschreibt entsprechende [bewährte Methoden und Richtlinien](#).

- **Knotentypen** — EC2 Amazon-Instances in einem EMR-Cluster sind in Knotentypen unterteilt. Es gibt drei Knotentypen: Primärknoten, Core-Knoten und Aufgabenknoten. Jeder Knotentyp führt eine Reihe von Rollen aus, die durch die von Ihnen auf dem Cluster installierten verteilten Anwendungen definiert werden. Während eines Hadoop MapReduce - oder Spark-Jobs verarbeiten Komponenten auf Kern- und Taskknoten beispielsweise Daten, übertragen die Ausgabe an Amazon S3 oder HDFS und stellen Statusmetadaten zurück an den primären Knoten. Bei einem einzigen Knoten-Cluster werden alle Komponenten auf dem Primärknoten ausgeführt. Weitere Informationen finden Sie unter [Verstehen Sie die Knotentypen in Amazon EMR: Primär-, Kern- und Task-Knoten](#).
- **EC2 Instances** — Wenn Sie einen Cluster erstellen, treffen Sie Entscheidungen über die EC2 Amazon-Instances, auf denen die einzelnen Knotentypen ausgeführt werden sollen. Der EC2 Instance-Typ bestimmt das Verarbeitungs- und Speicherprofil des Knotens. Die Wahl der EC2 Amazon-Instance für Ihre Knoten ist wichtig, da sie das Leistungsprofil der einzelnen Knotentypen in Ihrem Cluster bestimmt. Weitere Informationen finden Sie unter [EC2 Amazon-Instance-Typen für die Verwendung mit Amazon EMR konfigurieren](#).
- **Netzwerk** – Sie können Ihren Amazon-EMR-Cluster in einer VPC starten, indem Sie ein öffentliches Subnetz, ein privates Subnetz oder ein gemeinsam genutztes Subnetz verwenden. Ihre Netzwerkkonfiguration bestimmt, wie Kunden und Services Verbindungen zu Clustern herstellen können, um ihre Arbeit zu erledigen, wie Cluster mit Datenspeichern und anderen AWS -Ressourcen verbunden werden und welche Optionen Sie zur Steuerung des Datenverkehrs auf diesen Verbindungen haben. Weitere Informationen finden Sie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#).
- **Instance-Gruppierung** — Die Sammlung von EC2 Instances, die jeden Knotentyp hosten, wird entweder als Instance-Flotte oder als einheitliche Instance-Gruppe bezeichnet. Die Konfiguration der Instance-Gruppierung ist eine Auswahl, die Sie beim Erstellen eines Clusters treffen. Diese Auswahl bestimmt, wie Sie Ihrem Cluster Knoten hinzufügen können, während er läuft. Die Konfiguration gilt für alle Knotentypen. Er kann später nicht mehr geändert werden. Weitere Informationen finden Sie unter [Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen](#).

 Note

Die Konfiguration der Instance-Flotten ist nur in den Amazon-EMR-Versionen 4.8.0 und höher verfügbar, mit Ausnahme von 5.0.0 und 5.0.3.

Verstehen Sie die Knotentypen in Amazon EMR: Primär-, Kern- und Task-Knoten

In diesem Abschnitt erfahren Sie, wie Amazon EMR die einzelnen Knotentypen jeweils verwendet. Damit lernen Sie die Grundsätze der Kapazitätsplanung für Cluster kennen.

Primärknoten

Der Primärknoten verwaltet die Cluster und führt die Master-Komponenten von verteilten Anwendungen aus. Auf dem primären Knoten wird beispielsweise der ResourceManager YARN-Service ausgeführt, um Ressourcen für Anwendungen zu verwalten. Er führt auch den NameNode HDFS-Dienst aus, verfolgt den Status der an den Cluster übermittelten Jobs und überwacht den Zustand der Instanzgruppen.

Um den Fortschritt eines Clusters zu überwachen und direkt mit Anwendungen zu interagieren, können Sie über SSH eine Verbindung mit dem Primärknoten als Hadoop-Benutzer herstellen. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#). Durch das Verbinden mit dem Primärknoten erhalten Sie direkten Zugriff auf Verzeichnisse und Dateien, wie z. B. Hadoop-Protokolldateien. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#). Sie können auch Benutzeroberflächen anzeigen, die von den Anwendungen als auf dem Primärknoten ausgeführte Websites veröffentlicht werden. Weitere Informationen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

 Note

Mit Amazon EMR 5.23.0 und höher können Sie einen Cluster mit drei primären Knoten starten, um die Hochverfügbarkeit von Anwendungen wie YARN Resource Manager, HDFS, Spark NameNode, Hive und Ganglia zu unterstützen. Der Primärknoten ist mit diesem Feature keine potenzielle einzelne Fehlerquelle mehr. Wenn ein Primärknoten ausfällt, führt Amazon EMR automatisch einen Failover zu einem Standby-Primärknoten aus und

ersetzt den ausgefallenen Primärknoten durch einen neuen Primärknoten mit der gleichen Konfiguration und den gleichen Bootstrap-Aktionen. Weitere Informationen finden Sie unter [Primärknoten planen und konfigurieren](#).

Core-Knoten

Core-Knoten werden vom Primärknoten verwaltet. Core-Knoten führen den DataNode-Daemon zum Koordinieren der Datenspeicherung im Rahmen des Hadoop Distributed File System (HDFS) aus. Außerdem führen sie den TaskTracker-Daemon und andere parallele Rechenaufgaben für Daten aus, die für installierte Anwendungen erforderlich sind. Auf einem Core-Knoten werden beispielsweise NodeManager YARN-Daemons, Hadoop-Aufgaben und Spark-Executors ausgeführt. MapReduce

Es gibt nur eine Core-Instance-Gruppe oder Instance-Flotte pro Cluster, aber es können mehrere Knoten auf mehreren EC2 Amazon-Instances in der Instance-Gruppe oder Instance-Flotte laufen. Mit Instanzgruppen können Sie EC2 Amazon-Instances hinzufügen und entfernen, während der Cluster läuft. Sie können auch ein Auto Scaling einrichten, um Instances auf der Grundlage des Werts einer Metrik hinzuzufügen. Weitere Informationen zum Hinzufügen und Entfernen von EC2 Amazon-Instances mit der Instanzgruppen-Konfiguration finden Sie unter [Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen](#).

Mit Instance-Flotten können Sie Instances effektiv hinzufügen und entfernen, indem Sie die Zielkapazitäten der Instance-Flotte für On-Demand- und Spot Instances entsprechend anpassen. Weitere Informationen zu den Zielkapazitäten finden Sie unter [Instance-Flotten-Optionen](#).

Warning

Das Entfernen von HDFS-Daemons aus einem Core-Knoten, der ausgeführt wird, oder das Beenden von Core-Knoten können zu Datenverlusten führen. Seien Sie beim Konfigurieren von Core-Knoten für die Verwendung von Spot Instances vorsichtig. Weitere Informationen finden Sie unter [Wann sollten Sie Spot Instances verwenden?](#).

Aufgabenknoten

Sie können Task-Knoten verwenden, um Leistung für parallel Berechnungsaufgaben für Daten hinzuzufügen, z. B. MapReduce Hadoop-Aufgaben und Spark-Executoren. Aufgabenknoten

führen weder den DataNode-Daemon aus noch speichern sie Daten in HDFS. Wie bei Core-Nodes können Sie Task-Knoten zu einem Cluster hinzufügen, indem Sie EC2 Amazon-Instances zu einer bestehenden einheitlichen Instance-Gruppe hinzufügen oder indem Sie die Zielkapazitäten für eine Task-Instance-Flotte ändern.

Mit der einheitlichen Instance-Gruppenkonfiguration können Sie über bis zu 48 Aufgaben-Instance-Gruppen verfügen. Durch die Möglichkeit, Instance-Gruppen auf diese Weise hinzuzufügen, können Sie EC2 Amazon-Instance-Typen und Preisoptionen wie On-Demand-Instances und Spot-Instances kombinieren. Dadurch haben Sie die Flexibilität, kosteneffizient auf Workload-Anforderungen zu reagieren.

Mit der Instance-Flottenkonfiguration ist die Möglichkeit integriert, Instance-Typen und Kaufoptionen zu kombinieren, sodass nur eine Aufgaben-Instance-Flotte vorhanden ist.

Da Spot Instances häufig zum Ausführen von Aufgabenknoten verwendet werden, verfügt Amazon EMR über Standardfunktionen für die Planung von YARN-Aufträge, sodass laufende Aufträge nicht fehlschlagen, wenn Aufgabenknoten, die auf Spot Instances ausgeführt werden, beendet werden. Amazon EMR ermöglicht dies, indem Anwendungsmasterprozesse nur auf Core-Knoten ausgeführt werden können. Der Anwendungsmasterprozess steuert die Ausführung von Aufträgen und muss während der gesamten Laufzeit des Auftrags aktiv bleiben.

Amazon-EMR-Version 5.19.0 und höher verwendet zu diesem Zweck das integrierte [YARN-Knotenbeschriftungsfeature](#). (Frühere Versionen verwendeten einen Code-Patch). Die Eigenschaften in den Klassifizierungen `yarn-site` und in der `capacity-scheduler`-Konfiguration sind standardmäßig so konfiguriert, dass der YARN-Kapazitätsplaner und der Fair-Scheduler die Vorteile von Knotenbezeichnungen nutzen. Amazon EMR kennzeichnet Core-Knoten automatisch mit dem CORE-Label und legt Eigenschaften fest, sodass Anwendungsmaster nur für Knoten mit dem CORE-Label geplant werden. Durch manuelles Ändern verwandter Eigenschaften in den Konfigurationsklassifizierungen von Yarn-Site und Kapazitätsplaner oder direkt in den zugehörigen XML-Dateien könnte diese Feature beeinträchtigt oder verändert werden.

Beginnend mit der Amazon-EMR-6.x-Release-Reihe ist das Feature YARN-Knotenbeschriftungen standardmäßig deaktiviert. Die Anwendungs-Primär-Prozesse können standardmäßig sowohl auf Core- als auch auf Aufgabenknoten ausgeführt werden. Sie können die Funktion für YARN-Knotenbeschriftungen aktivieren, indem Sie folgende Eigenschaften konfigurieren:

- `yarn.node-labels.enabled: true`
- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

Beginnend mit der Amazon EMR 7.x-Release-Serie weist Amazon EMR Instances YARN-Knotenbezeichnungen nach ihrem Markttyp zu, z. B. On-Demand oder Spot. Sie können Knotenlabels aktivieren und Anwendungsprozesse auf ON_DEMAND beschränken, indem Sie die folgenden Eigenschaften konfigurieren:

```
yarn.node-labels.enabled: true
yarn.node-labels.am.default-node-label-expression: 'ON_DEMAND'
```

Wenn Sie Amazon EMR 7.0 oder höher verwenden, können Sie den Anwendungsprozess mit der folgenden Konfiguration auf Knoten mit dem CORE Label beschränken:

```
yarn.node-labels.enabled: true
yarn.node-labels.am.default-node-label-expression: 'CORE'
```

Für Amazon EMR-Versionen 7.2 und höher gilt: Wenn Ihr Cluster verwaltete Skalierung mit Knotenbezeichnungen verwendet, versucht Amazon EMR, den Cluster auf der Grundlage des Anwendungsprozesses und der Nachfrage der Executoren unabhängig voneinander zu skalieren.

Wenn Sie beispielsweise Amazon EMR-Versionen 7.2 oder höher verwenden und den Anwendungsprozess auf Knoten beschränken, skaliert Managed Scaling die ON_DEMAND Knoten nach oben ON_DEMAND, wenn die Nachfrage nach Anwendungsprozessen steigt. Wenn Sie den Anwendungsprozess auf Knoten beschränken, werden bei der verwalteten Skalierung die CORE Knoten entsprechend CORE hochskaliert, wenn die Nachfrage nach Anwendungsprozessen steigt.

Informationen zu spezifischen Eigenschaften finden Sie unter [Amazon-EMR-Einstellungen, die Aufgabenfehler aufgrund des Beendens von Aufgabenknoten-Spot Instances verhindern](#).

EC2 Amazon-Instance-Typen für die Verwendung mit Amazon EMR konfigurieren

EC2 Instances gibt es in verschiedenen Konfigurationen, die als Instance-Typen bezeichnet werden. Instance-Typen verfügen über andere CPU-, Eingabe/Ausgabe- und Speicherkapazitäten. Zusätzlich zum Instance-Typ können Sie verschiedene Kaufoptionen für EC2 Amazon-Instances wählen. Sie können verschiedene Instance-Typen und Kaufoptionen innerhalb von einheitlichen Instance-Gruppen oder Instance-Flotten angeben. Weitere Informationen finden Sie unter [Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen](#). Hinweise zur Auswahl von Instance-Typen und Kaufoptionen für Ihre Anwendung finden Sie unter [Konfiguration von Amazon EMR-Cluster-Instance-Typen und Best Practices für Spot-Instances](#).

 Important

Wenn Sie mithilfe von einem Instance-Typ auswählen AWS Management Console, entspricht die Anzahl der vCPUs, die für jeden Instance-Typ angezeigt wird, der Anzahl der YARN-vcores für diesen Instance-Typ, nicht der Anzahl von EC2 v CPUs für diesen Instance-Typ. Weitere Informationen zur Anzahl von v CPUs für jeden Instance-Typ finden Sie unter [EC2 Amazon-Instance-Typen](#).

Themen

- [Unterstützte Instance-Typen mit Amazon EMR](#)
- [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#)
- [Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen](#)

Unterstützte Instance-Typen mit Amazon EMR

In diesem Abschnitt werden die Instance-Typen beschrieben, die Amazon EMR unterstützt, geordnet nach AWS-Region. Weitere Informationen zu Instance-Typen finden Sie unter [EC2 Amazon-Instances](#) und [Amazon Linux AMI-Instance-Typmatrix](#).

Nicht alle Instance-Typen sind in allen Regionen verfügbar. Die Instance-Verfügbarkeit hängt von der Verfügbarkeit und der Nachfrage in der angegebenen Region und Availability Zone ab. Die Availability Zone einer Instance wird durch das Subnetz bestimmt, das Sie zum Starten Ihres Clusters verwenden.

Überlegungen

Berücksichtigen Sie Folgendes, wenn Sie Instance-Typen für Ihren Amazon-EMR-Cluster auswählen.

 Important

Wenn Sie mithilfe von einem Instance-Typ auswählen AWS Management Console, entspricht die Anzahl der vCPUs, die für jeden Instance-Typ angezeigt wird, der Anzahl der YARN-vcores für diesen Instance-Typ, nicht der Anzahl von EC2 v CPUs für diesen Instance-Typ. Weitere Informationen zur Anzahl von v CPUs für jeden Instance-Typ finden Sie unter [EC2 Amazon-Instance-Typen](#).

- Wenn Sie einen Cluster mit einem Instance-Typ erstellen, der in der angegebenen Region und Verfügbarkeitszone nicht verfügbar ist, schlägt die Bereitstellung Ihres Clusters möglicherweise fehl oder die Bereitstellung bleibt hängen. Informationen zur Instance-Verfügbarkeit finden Sie auf der [Amazon-EMR-Preisseite](#) oder in den [Unterstützte Instance-Typen von AWS-Region](#)-Tabellen auf dieser Seite.
- Ab Amazon-EMR-Version 5.13.0 verwenden alle Instances HVM-Virtualisierung und EBS-gestützten Speicher für Stamm-Volumes. Bei der Verwendung von Amazon-EMR-Versionen vor Version 5.13.0 nutzen einige Instances der vorherigen Generation PVM-Virtualisierung. Weitere Informationen finden Sie unter [Linux AMI-Virtualisierungstypen](#).
- Aufgrund mangelnder Hardwareunterstützung und fehlender Standardeinstellungen, die zu einer Unterauslastung von Arbeitsspeicher und Kernen führen können, empfehlen wir Ihnen nicht, die Instance-Typen,,,,,, zu verwenden,,,,,,c7a,,,,,,c7i,,m7i,,m7i-flex,,r7a,,r7i,,r7iz,,i4i.12xlarge,,,, verwenden, i4i.24xlarge wenn Sie Amazon EMR-Versionen unter 5.36.1 und 6.10.0 ausführen. Wenn Sie diese Instance-Typen in diesen Versionen ausführen, kann es zu Leistungseinbußen kommen und Sie werden nicht die erwarteten Vorteile neuerer Instance-Typen wie vs sehen. c7i c6i Für eine optimale Ressourcennutzung und Leistung bei diesen Leistungstypen sollten Sie 5.36.1 und höher oder 6.10.0 und höher ausführen, um deren Funktionen zu maximieren.
- Einige Instance-Typen unterstützen Enhanced Networking. Weitere Informationen finden Sie unter [Enhanced Networking in Linux](#).
- NVIDIA- und CUDA-Treiber sind auf GPU-Instance-Typen standardmäßig installiert.

Unterstützte Instance-Typen von AWS-Region

In den folgenden Tabellen sind die EC2 Amazon-Instance-Typen aufgeführt, die Amazon EMR unterstützt, geordnet nach AWS-Region. In den Tabellen sind auch die frühesten Amazon-EMR-Versionen der Serien 5.x, 6.x und 7.x aufgeführt, die die einzelnen Instance-Typen unterstützen.

USA Ost (Nord-Virginia) – us-east-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Beschleunigte Datenverarbeitung	f2.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i7i.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7i.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

USA Ost (Ohio) – us-east-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7a.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.6.0, emr-5.0.1, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	p5.48xlarge	emr-6.14.0, emr-7.0.0
	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7a.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7iz.xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.1, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7i.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

USA West (Nordkalifornien) – us-west-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

USA West (Oregon) – us-west-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Beschleunigte Datenverarbeitung	f2.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	f2.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i7i.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7i.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

AWS GovCloud (US-West) — -1 us-gov-west

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Beschleunigte Datenverarbeitung	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Speicheroptimiert	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

AWS GovCloud (US-Ost) - -1 us-gov-east

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Afrika (Kapstadt) – af-south-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Hongkong) – ap-east-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Jakarta) – ap-southeast-3

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m6g.xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.2xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.4xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.8xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.16xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i.48xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i-flex.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i-flex.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i-flex.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m7i-flex.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c6g.xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.2xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.4xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.8xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.12xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.16xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r6g.xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.2xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.4xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.8xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.12xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.16xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r7i.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.48xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.6xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Melbourne) - ap-southeast-4

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i.48xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i-flex.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i-flex.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i-flex.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i-flex.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i-flex.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	m7i-flex.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r5.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	r7i.48xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.4xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.8.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Malaysia) - ap-southeast-5

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m6g.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6g.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6g.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6g.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6g.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6gd.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6gd.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6gd.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6gd.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6gd.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6gd.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6i.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m6id.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7g.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7g.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7g.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7g.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7g.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7gd.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7gd.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7gd.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7gd.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7gd.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7gd.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i.48xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i-flex.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i-flex.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i-flex.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i-flex.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m7i-flex.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	m7i-flex.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6g.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6g.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6g.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6g.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6g.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6g.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6gn.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6gn.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6gn.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6gn.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6gn.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6i.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6id.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c6in.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7g.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7g.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7g.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7g.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7g.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7g.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7gd.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7gd.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7gd.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7gd.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7gd.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i.48xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i-flex.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i-flex.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Beschleunigte Datenverarbeitung	c7i-flex.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i-flex.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i-flex.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	c7i-flex.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
RAM-optimiert	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	r6g.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6g.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6g.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6g.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6g.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6i.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r6id.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7g.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7g.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7g.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7g.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7g.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7gd.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7gd.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7gd.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7gd.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7gd.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7gd.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	r7i.48xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2idn.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2idn.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2idn.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2iedn.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Speicheroptimiert	x2iedn.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i3en.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i3en.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i3en.3xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i3en.6xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i3en.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i3en.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.2xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.4xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.8xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.12xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.16xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.24xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0
	i4i.32xlarge	emr-5.36.2, emr-6.11.1, emr-7.0.0

Asien-Pazifik (Mumbai) – ap-south-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i-flex.16xlarge	emr-4.6.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asien-Pazifik (Hyderabad) – ap-south-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Osaka) – ap-northeast-3

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Seoul) – ap-northeast-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Asien-Pazifik (Singapur) – ap-southeast-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asien-Pazifik (Sydney) – ap-southeast-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	f2.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asien-Pazifik (Tokio) – ap-northeast-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kanada (Zentral) – ca-central-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m7i-flex.12xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.8.2, emr-5.0.2, emr-6.0.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.48xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.2, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kanada West (Calgary) – ca-west-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m6id.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.9xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.18xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Speicheroptimiert	i3en.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.6xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

China (Ningxia) – cn-northwest-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

China (Peking) – cn-north-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Europa (Frankfurt) – eu-central-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	m8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	c8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0
	g6e.4xlarge	emr-7.5.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8gd.xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.2xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.4xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.8xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.12xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.16xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.24xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	r8gd.48xlarge	emr-5.36.2, emr-6.15.0, emr-7.1.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europa (Zürich) – eu-central-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
RAM-optimiert	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Europa (Irland) – eu-west-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7a.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7iz.32xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europa (London) – eu-west-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.8.2, emr-5.0.3, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	f2.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	f2.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.4xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.3, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i7ie.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i7ie.3xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.6xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.18xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i7ie.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europa (Mailand) – eu-south-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Europa (Spanien) – eu-south-2

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7a.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7a.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7i-flex.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7a.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7a.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i-flex.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i-flex.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i-flex.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i-flex.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i-flex.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c7i-flex.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
RAM-optimiert	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7a.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7a.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7a.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7i.48xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	im4gn.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	im4gn.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	im4gn.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	im4gn.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	im4gn.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Europa (Paris) – eu-west-3

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.16xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.8xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.9.2, emr-5.5.3, emr-6.0.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.16xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.5.3, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europa (Stockholm) – eu-north-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7a.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.32xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7a.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i-flex.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7a.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7a.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.32xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7a.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i-flex.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g6e.xlarge	emr-7.5.0
	g6e.2xlarge	emr-7.5.0
	g6e.4xlarge	emr-7.5.0
	g6e.8xlarge	emr-7.5.0
	g6e.12xlarge	emr-7.5.0
	g6e.16xlarge	emr-7.5.0
	g6e.24xlarge	emr-7.5.0
	g6e.48xlarge	emr-7.5.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7a.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7a.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.32xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7a.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r7i.48xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Israel (Tel Aviv) - il-central-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gn.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gn.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gn.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gn.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6gn.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Beschleunigte Datenverarbeitung	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	d3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	d3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Naher Osten (Bahrain) – me-south-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Naher Osten (VAE) – me-central-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Für Datenverarbeitung optimiert	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
RAM-optimiert	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Südamerika (São Paulo) – sa-east-1

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
Allgemeine Zwecke	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	m7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	m7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
Für Datenverarbeitung optimiert	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c6in.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.2xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.4xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	c7i-flex.8xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.12xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c7i-flex.16xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	c8g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.24xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c8g.48xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Beschleunigte Datenverarbeitung	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g6.xlarge	emr-7.2.0
	g6.2xlarge	emr-7.2.0
	g6.4xlarge	emr-7.2.0
	g6.8xlarge	emr-7.2.0
	g6.12xlarge	emr-7.2.0
	g6.16xlarge	emr-7.2.0
	g6.24xlarge	emr-7.2.0
	g6.48xlarge	emr-7.2.0
	gr6.4xlarge	emr-7.2.0
	gr6.8xlarge	emr-7.2.0
	p4d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
RAM-optimiert	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.2xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.4xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.8xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.16xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	r7i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	r7i.48xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Speicheroptimiert	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instance-Klasse	Instance-Typ	Unterstützte Mindestversion von Amazon EMR (5.x, 6.x, 7.x)
	i4i.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.12xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	i4i.24xlarge	emr-5.0.0, emr-6.0.0, emr-7.0.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Instances der vorherigen Generation

Amazon EMR unterstützt Instances der vorherigen Generation zur Unterstützung von Anwendungen, die für diese Instances optimiert sind und noch nicht aktualisiert wurden. Weitere Informationen zu diesen Instance-Typen und Upgrade-Pfaden finden Sie unter [Instances der vorherigen Generation](#).

Instance-Klasse	Instance-Typen
General Purpose	m1.small ¹ m1.medium ¹ m1.large ¹ m1.xlarge ¹ m3.xlarge ¹ m3.2xlarge ¹ m4.large m4.xlarge m4.2xlarge m4.4xlarge m4.10xlarge m4.16xlarge

Instance-Klasse	Instance-Typen
Compute Optimized	c1.medium ^{1 2} c1.xlarge ¹ c3.xlarge ¹ c3.2xlarge ¹ c3.4xlarge ¹ c3.8xlarge ¹ c4.large c4.xlarge c4.2xlarge c4.4xlarge c4.8xlarge
Memory Optimized	m2.xlarge ¹ m2.2xlarge ¹ m2.4xlarge ¹ r3.xlarge r3.2xlarge r3.4xlarge r3.8xlarge r4.xlarge r4.2xlarge r4.4xlarge r4.8xlarge r4.16xlarge
Storage Optimized	d2.xlarge d2.2xlarge d2.4xlarge d2.8xlarge i2.xlarge i2.2xlarge i2.4xlarge i2.8xlarge

¹ Verwendet das PVM-Virtualisierungs-AMI mit Amazon-EMR-Versionen vor 5.13.0. Weitere Informationen finden Sie unter [Linux-AMI-Virtualisierungstypen](#).

² Nicht unterstützt in Version 5.15.0.

Kaufoptionen für Instanzen in Amazon EMR

Wenn Sie einen Cluster einrichten, wählen Sie eine Kaufoption für EC2 Amazon-Instances. Sie können On-Demand-Instances, Spot Instances oder beides auswählen. Die Preise variieren basierend auf dem Instance-Typ und der Region. Der Amazon EMR-Preis wird zusätzlich zum EC2 Amazon-Preis (der Preis für die zugrunde liegenden Server) und zum Amazon EBS-Preis (wenn Amazon EBS-Volumes angehängt werden) berechnet. Aktuelle Preisangaben finden Sie unter [Amazon-EMR-Preise](#).

Ihre Wahl zur Verwendung von Instance-Gruppen oder Instance-Flotten in Ihrem Cluster bestimmt, wie Sie die Instance-Kaufoptionen ändern können, während der Cluster ausgeführt wird. Wenn Sie einheitliche Instance-Gruppen wählen, können Sie die Kaufoption für eine Instance-Gruppe nur angeben, wenn Sie sie erstellen. Der Instance-Typ und die Kaufoption gelten für alle EC2 Amazon-Instances in jeder Instance-Gruppe. Bei der Wahl von Instance-Flotten können Sie die Kaufoptionen ändern, nachdem eine Instance-Flotte erstellt wurde. Sie können die Kaufoptionen kombinieren, um eine festgelegte Zielkapazität zu erfüllen. Weitere Informationen zu diesen Konfigurationen finden Sie unter [Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen](#).

On-Demand Instances

Bei On-Demand-Instances zahlen Sie für die Rechenkapazität nach Sekunde. Optional können Sie für diese On-Demand-Instances Reserved Instance- oder Dedicated Instance-Kaufoptionen verwenden. Bei Reserved Instances leisten Sie eine einmalige Zahlung für eine Instance, um Kapazität zu reservieren. Dedicated Instances sind auf Host-Hardwareebene physisch von Instances isoliert, die zu anderen AWS Konten gehören. Weitere Informationen zu Kaufoptionen finden Sie unter [Instance-Kaufoptionen](#) im EC2 Amazon-Benutzerhandbuch.

Verwenden von Reserved Instances

Um Reserved Instances in Amazon EMR EC2 zu verwenden, kaufen Sie die Reserved Instance über Amazon und geben die Reservierungsparameter an, einschließlich des Umfangs der Reservierung, der entweder für eine Region oder eine Availability Zone gilt. Weitere Informationen finden Sie unter [Amazon EC2 Reserved Instances](#) und [Buying Reserved Instances](#) im EC2 Amazon-Benutzerhandbuch. Nachdem Sie eine Reserved Instance gekauft haben, verwendet Amazon EMR die Reserved Instance, wenn ein Cluster startet und alle der folgenden Bedingungen erfüllt sind:

- Eine On-Demand-Instance ist in der Cluster-Konfiguration angegeben, die mit der Reserved-Instance-Spezifikation übereinstimmt.
- Der Cluster wird im Rahmen der Instance-Reservierung (Availability Zone oder Region) gestartet.
- Die Reserved Instance-Kapazität ist noch verfügbar.

Angenommen, Sie kaufen eine Reserved Instance `m5.xlarge` mit der gewünschten Instance-Reservierung für die Region USA Ost. Anschließend starten Sie einen Amazon-EMR-Cluster in USA Ost, der zwei `m5.xlarge`-Instances verwendet. Die erste Instance wird nach dem Tarif für Reserved Instances abgerechnet und die andere nach dem On-Demand-Tarif. Die Reserved Instance-Kapazität wird verwendet, bevor die On-Demand-Instances erstellt werden.

Verwenden von Dedicated Instances

Um Dedicated Instances zu verwenden, kaufen Sie Dedicated Instances über Amazon EC2 und erstellen dann eine VPC mit dem Dedicated Tenancy-Attribut. Anschließend geben Sie in Amazon EMR an, dass ein Cluster in dieser VPC gestartet werden soll. Alle On-Demand-Instances im Cluster, die der Dedicated Instance-Spezifikation entsprechen, verwenden verfügbare Dedicated Instances beim Start des Clusters.

 Note

Amazon EMR unterstützt das `dedicated` Attribut für einzelne Instances nicht.

Spot Instances

Spot-Instances in Amazon EMR bieten Ihnen die Möglichkeit, EC2 Amazon-Instance-Kapazität zu geringeren Kosten als beim Kauf auf Abruf zu erwerben. Der Nachteil der Verwendung von Spot Instances besteht darin, dass Instances möglicherweise beendet werden, wenn die Spot-Kapazität für den von Ihnen ausgeführten Instance-Typ nicht mehr verfügbar ist. Weitere Informationen dazu, wann Sie Spot Instances für Ihre Anwendung verwenden sollten, finden Sie unter [Wann sollten Sie Spot Instances verwenden?](#).

Wenn Amazon EC2 über ungenutzte Kapazitäten verfügt, bietet Amazon EC2 Instances zu reduzierten Kosten an, die als Spot-Preis bezeichnet werden. Dieser Preis schwankt abhängig von Verfügbarkeit und Bedarf und wird nach Region und Availability Zone festgelegt. Wenn Sie sich für Spot-Instances entscheiden, geben Sie den maximalen Spot-Preis an, den Sie für jeden EC2 Instance-Typ zu zahlen bereit sind. Wenn der Spot-Preis in der Availability Zone des Clusters unter dem für diesen Instance-Typ angegebenen maximalen Spot-Preis liegt, werden Instances gestartet. Während die Instances ausgeführt werden, wird Ihnen der aktuelle Spot-Preis nicht Ihr maximaler Spot-Preis in Rechnung gestellt.

 Note

Spot-Instances mit definierter Laufzeit (auch Spot-Blöcke genannt) stehen Neukunden ab dem 1. Juli 2021 nicht mehr zur Verfügung. Für Kunden, die diese Funktion bereits genutzt haben, werden wir Spot-Instances mit einer definierten Laufzeit bis zum 31. Dezember 2022 weiterhin unterstützen.

Aktuelle Preise finden Sie unter Preise für [Amazon EC2 Spot-Instances](#). Weitere Informationen finden Sie unter [Spot-Instances](#) im EC2 Amazon-Benutzerhandbuch. Wenn Sie einen Cluster erstellen und konfigurieren, geben Sie Netzwerkoptionen an, die letztendlich die Availability Zone bestimmen, in der Ihr Cluster gestartet wird. Weitere Informationen finden Sie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#).

 Tip

Sie können den aktuellen Spot-Preis in der Konsole anzeigen, indem Sie mit dem Mauszeiger auf die QuickInfo für Informationen neben der Kaufoption Spot (Spot) zeigen, wenn Sie einen Cluster mittels Advanced Options (Erweiterte Optionen) erstellen. Die Preise für jede Availability Zone in der ausgewählten Region werden angezeigt. Die grünen Zeilen enthalten die niedrigsten Preise. Aufgrund der Spot-Preisschwankungen zwischen Availability Zones kann es sein, dass durch Auswählen der Availability Zone mit dem niedrigsten Anfangspreis möglicherweise nicht der niedrigste Preis für die Nutzungsdauer des Clusters erzielt wird. Um optimale Ergebnisse zu erzielen, sehen Sie sich den Availability Zone-Preisverlauf an, bevor Sie sich entscheiden. Weitere Informationen finden Sie unter [Preisverlauf für Spot-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Die Spot-Instance-Optionen hängen davon ab, ob Sie einheitliche Instance-Gruppen oder Instance-Flotten in der Cluster-Konfiguration verwenden.

Spot-Instances in einheitlichen Instance-Gruppen

Wenn Sie Spot-Instances in einer einheitlichen Instance-Gruppe verwenden, muss es sich bei allen Instances in der Instance-Gruppe um Spot-Instances handeln. Sie geben ein einzelnes Subnetz eine oder Availability Zone für den Cluster an. Für jede Instance-Gruppe legen Sie eine einzelne Spot Instance und einen maximalen Spot-Preis fest. Die Spot Instances des entsprechenden Typs werden gestartet, wenn der Spot-Preis in der Region und Availability Zone des Clusters unter dem maximalen Spot-Preis liegt. Instances werden beendet, wenn der Spot-Preis Ihren maximalen Spot-Preis übersteigt. Sie legen den maximalen Spot-Preis nur beim Konfigurieren einer Instance-Gruppe fest. Er kann später nicht mehr geändert werden. Weitere Informationen finden Sie unter [Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen](#).

Spot-Instances in Instance-Flotten

Wenn Sie die Instance-Flottenkonfiguration verwenden, erhalten Sie durch zusätzliche Optionen mehr Kontrolle darüber, wie Spot-Instances gestartet und beendet werden. Grundsätzlich verwenden Instance-Flotten zum Starten von Instances eine andere Methode als einheitliche Instance-Gruppen. Hierbei legen Sie eine Zielkapazität für Spot-Instances (und On-Demand-Instances) und bis zu fünf Instance-Typen fest. Sie können auch eine gewichtete Kapazität für jeden Instance-Typ festlegen oder die vCPU (YARN vcores) des Instance-Typs als gewichtete Kapazität verwenden. Die gewichtete Kapazität wird im Rahmen der Zielkapazität berücksichtigt, wenn eine Instance

dieses Typs bereitgestellt wird. Amazon EMR stellt Instances mit beiden Kaufoptionen bereit, bis die Zielkapazität für jedes Ziel erfüllt ist. Darüber hinaus können Sie eine Vielzahl von Availability Zones definieren, die Amazon EMR zum Starten von Instances zur Wahl stehen. Sie stellen außerdem zusätzliche Spot-Optionen für jede Flotte bereit, einschließlich eines Bereitstellungs-Timeouts. Weitere Informationen finden Sie unter [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#).

Speicheroptionen und Verhalten von Instances in Amazon EMR

Übersicht

Der Instance-Speicher und der Amazon-EBS-Volume-Speicher werden für HDFS-Daten sowie für Puffer, Caches, Arbeitsdaten und andere temporäre Inhalte verwendet, die einige Anwendungen möglicherweise in das lokale Dateisystem „verschütten“.

Amazon EBS funktioniert innerhalb von Amazon EMR anders als bei regulären EC2 Amazon-Instances. An Amazon-EMR-Cluster angefügte Amazon-EMR-Volumes sind beispielsweise flüchtig: Die Volumes werden beim Beenden des Clusters und der Instance gelöscht (z. B. beim Verkleinern von Instance-Gruppen). Daher sollten Sie nicht davon ausgehen, dass Daten dauerhaft gespeichert werden. Obwohl die Daten flüchtig sind, ist es möglich, dass Daten in HDFS abhängig von der Anzahl und der Spezialisierung der Knoten im Cluster repliziert werden. Wenn Sie Amazon-EBS-Speichervolumes hinzufügen, werden diese als zusätzliche Volumes bereitgestellt. Sie sind nicht Teil des Startvolumes. YARN ist so konfiguriert, dass alle zusätzlichen Volumes verwendet werden. Sie sind jedoch dafür verantwortlich, die zusätzlichen Volumes als lokalen Speicher (z. B. für lokale Protokolldateien) zuzuweisen.

Überlegungen

Die folgenden zusätzlichen Überlegungen sollten Sie berücksichtigen, wenn Sie Amazon EBS mit EMR-Clustern verwenden:

- Sie können nicht einen Snapshot eines Amazon-EBS-Volumes erstellen und dann innerhalb von Amazon EMR wiederherstellen. Um wiederverwendbare benutzerdefinierte Konfigurationen zu erstellen, verwenden Sie ein benutzerdefiniertes AMI (verfügbar ab Version 5.7.0 von Amazon EMR). Weitere Informationen finden Sie unter [Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration](#).
- Ein verschlüsseltes Amazon-EBS-Root-Volume wird nur unterstützt, wenn Sie ein benutzerdefiniertes AMI verwenden. Weitere Informationen finden Sie unter [Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Amazon-EBS-Root-Gerät-Datenträger](#).

- Wenn Sie Tags mit der Amazon-EMR-Webservice-API zuweisen, werden diese Operationen auf EBS-Volumes angewendet.
- Es gilt eine Beschränkung von 25 Volumes pro Instance.
- Die Amazon-EBS-Volumes auf den Core-Knoten dürfen nicht weniger als 5 GB groß sein.
- Amazon EBS hat ein festes Limit von 2.500 EBS-Volumes pro Instance-Startanfrage. Dieses Limit gilt auch für Amazon EMR auf EC2 Clustern. Wir empfehlen, dass Sie Cluster mit der Gesamtzahl der EBS-Volumes innerhalb dieses Limits starten und den Cluster dann manuell oder mit Amazon EMR Managed Scaling nach Bedarf hochskalieren. [Weitere Informationen zum EBS-Volumenlimit finden Sie unter Servicekontingente.](#)

Amazon-EBS-Standardspeicher für Instances

Für EC2 Instances, die nur über EBS-Speicher verfügen, weist Amazon EMR Amazon EBS-gp2- oder gp3-Speicher-Volumes den Instances zu. Wenn Sie einen Cluster mit der Amazon-EMR-Version 5.22.0 und höher erstellen, erhöht sich die Standardmenge des Amazon-EBS-Speichers basierend auf der Größe der Instance.

Wir teilen jeden erhöhten Speicherplatz auf mehrere Volumes auf. Dies führt zu einer erhöhten IOPS-Leistung und damit bei einigen standardisierten Workloads zu einer höheren Leistung. Wenn Sie eine andere Instance-Speicherkonfiguration von Amazon EBS verwenden möchten, können Sie diese beim Erstellen eines EMR-Clusters bzw. beim Hinzufügen von Knoten zu einem Cluster angeben. Sie können gp2- oder gp3-Volumes von Amazon EBS als Root-Volumes verwenden und gp2- oder gp3-Volumes als zusätzliche Volumes hinzufügen. Weitere Informationen finden Sie unter [Angaben zusätzlicher EBS-Speicher-Volumes](#).

In der folgenden Tabelle sind die Standardanzahl von Amazon-EBS-GP2-Speicher-Volumes, Größen und Gesamtgrößen pro Instance-Typ aufgeführt. Hinweise zu gp2-Volumes im Vergleich zu gp3-Volumes finden Sie unter [Vergleichen der Amazon-EBS-Volumentypen gp2 und gp3](#).

Standardmäßige gp2-Speichervolumes von Amazon EBS und Speichergröße nach Instance-Typ für Amazon EMR 5.22.0 und höher

Instance-Größe	Anzahl der Volumes	Volume-Größe (GiB)	Gesamtgröße (GB)
*.large	1	32	32
*.xlarge	2	32	64

Instance-Größe	Anzahl der Volumes	Volume-Größe (GiB)	Gesamtgröße (GB)
*.2xlarge	4	32	128
*.4xlarge	4	64	256
*.8xlarge	4	128	512
9xlarge	4	144	576
10xlarge	4	160	640
12xlarge	4	192	768
*.16xlarge	4	256	1024
18xlarge	4	288	1 152
24xlarge	4	384	1536

Standard-Root-Volume von Amazon EBS für Instances

Mit Amazon EMR-Versionen 6.15 und höher hängt Amazon EMR automatisch eine Amazon EBS General Purpose SSD (gp3) als Root-Gerät an, um die Leistung zu verbessern. AMIs In früheren Versionen fügt Amazon EMR EBS-Allzweck-SSD (gp2) als Root-Gerät zu.

	6.15 und höher	6.14 und niedriger
Root-Volume-Standardtyp		
Standardgröße		
Standard-IOPS		
Standarddurchsatz		

Informationen zum Anpassen des Root-Geräte-Volumes von Amazon EBS finden Sie unter [Angeben zusätzlicher EBS-Speicher-Volumes](#).

Angeben zusätzlicher EBS-Speicher-Volumes

Wenn Sie Instance-Typen in Amazon EMR konfigurieren, können Sie zusätzliche EBS-Volumes angeben, um Kapazität über den Instance-Speicher (falls vorhanden) und das Standard-EBS-Volume hinaus hinzuzufügen. Amazon EBS bietet die folgenden Volume-Typen: Allzweck (SSD), Bereitgestellte IOPS (SSD), durchsatzoptimiert (HDD), Cold (HDD) und Magnetfestplatte. Diese unterscheiden sich bei den Leistungsmerkmalen und im Preis, sodass Sie Ihren Speicher den Analyse- und Business-Anforderungen Ihrer Anwendungen entsprechend anpassen können. Beispielsweise benötigen einige Anwendungen den Überlauf auf Datenträger, während andere im Speicher oder unter Verwendung Amazon S3 sicher arbeiten können.

Sie können Amazon-EBS-Volumes nur beim Cluster-Startup und beim Hinzufügen einer zusätzlichen Aufgabenknoten-Instance-Gruppe an Instances anhängen. Wenn eine Instance in einem Amazon-EMR-Cluster ausfällt, werden sowohl die Instance als auch die angeschlossenen Amazon-EBS-Volumes durch neue Volumes ersetzt. Wenn Sie ein Amazon-EBS-Volume manuell trennen, behandelt Amazon EMR dies als Fehler und ersetzt sowohl den Instance-Speicher (falls zutreffend) als auch die Volume-Speicher.

Amazon EMR erlaubt Ihnen nicht, Ihren Volumetyp für einen vorhandenen EMR-Cluster von gp2 auf gp3 zu ändern. Um gp3 für Ihre Workloads zu verwenden, starten Sie einen neuen EMR-Cluster. Darüber hinaus raten wir davon ab, den Durchsatz und die IOPS auf einem Cluster, der verwendet wird oder bereitgestellt wird, zu aktualisieren, da Amazon EMR für jede neue Instance, die bei der Hochskalierung des Clusters hinzugefügt wird, den Durchsatz und die IOPS-Werte verwendet, die Sie beim Clusterstart angegeben haben. Weitere Informationen erhalten Sie unter [Vergleichen der Amazon-EBS-Volumetypen gp2 und gp3](#) und [Auswahl von IOPS und Durchsatz bei der Migration zu gp3 Amazon EBS-Volumetypen](#).

Important

Um ein gp3-Volume mit Ihrem EMR-Cluster zu verwenden, müssen Sie einen neuen EMR-Cluster starten.

Vergleichen der Amazon-EBS-Volumetypen gp2 und gp3

Hier finden Sie einen Vergleich der Kosten zwischen den gp2- und gp3-Volumes in der Region USA Ost (Nord-Virginia). Die aktuellen Informationen finden Sie auf der Produktseite von [Allzweck-Volumes von Amazon EBS](#) und auf der [Amazon-EBS-Preisseite](#).

Volume-Typ	gp3	gp2
Volume-Größe	1 GiB–16 TiB	1 GiB–16 TiB
Standard-/Baseline-IOPS	3000	3 IOPS/GiB (mindestens 100 IOPS) bis maximal 16.000 IOPS. Volumes, die kleiner als 1 TiB sind, können auch bis zu 3 000 IOPS erreichen.
Max. IOPS pro Volume	16,000	16,000
Standard-/Baseline-Durchsatz	125 MiB/s	Die Durchsatzgrenze liegt je nach Volumengröße zwischen 128 MiB/s und 250 MiB/s.
Max. Durchsatz pro Volume	1 000 MiB/s	250 MiB/s
Preis	0,08 USD/GiB-Monat 3.000 IOPS kostenlos und 0,005 USD/bereitgestelltes IOPS-Monat über 3.000; 125 kostenlos und 0,04 USD/bereitgestellt MiB/s MiB/s-month over 125MiB/s	0,10 USD/GiB-Monat

Auswahl von IOPS und Durchsatz bei der Migration zu gp3 Amazon EBS-Volumetypen

Bei der Bereitstellung eines GP2-Volumes müssen Sie die Größe des Volumes ermitteln, um das Verhältnis zwischen IOPS und Durchsatz zu ermitteln. Mit gp3 müssen Sie kein größeres Volume bereitstellen, um eine höhere Leistung zu erzielen. Sie können die gewünschte Größe und Leistung je nach Anwendungsanforderungen wählen. Durch die Auswahl der richtigen Größe und der richtigen

Leistungsparameter (IOPS, Durchsatz) können Sie maximale Kostensenkungen erzielen, ohne die Leistung zu beeinträchtigen.

Die folgende Tabelle hilft Ihnen bei der Auswahl der gp3-Konfigurationsoptionen:

Volume-Größe	IOPS	Durchsatz
1–170 GiB	3000	125 MiB/s
170–334 GiB	3000	125 MiB/s , wenn der gewählte EC2 Instance-Typ 125 * unterstützt. MiB/s or less, use higher as per usage, Max 250 MiB/s
334–1 000 GiB	3000	125 MiB/s , wenn der gewählte EC2 Instance-Typ 125 MiB/s or less, Use higher as per usage, Max 250 MiB/s * unterstützt.
Über 1 000 GiB	Passen Sie gp2-IOPS (Größe in GiB x 3) oder maximale IOPS an, abhängig vom aktuellen gp2-Volume	125 MiB/s , wenn der gewählte EC2 Instance-Typ 125 MiB/s or less, Use higher as per usage, Max 250 MiB/s * unterstützt.

*Gp3 kann einen Durchsatz von bis zu 1000 MiB/s. Since gp2 provides a maximum of 250MiB/s bereitstellen. Wenn Sie gp3 verwenden, müssen Sie dieses Limit möglicherweise nicht überschreiten.

Konfiguration von Netzwerken in einer VPC für Amazon EMR

Die meisten Cluster werden mithilfe von Amazon Virtual Private Cloud (Amazon VPC) in einem virtuellen Netzwerk gestartet. Eine VPC ist ein isoliertes virtuelles Netzwerk innerhalb Ihres AWS Kontos AWS , das logisch isoliert ist. Sie können Aspekte wie private IP-Adressbereiche, Subnetze, Routing-Tabellen und Netzwerk-Gateways konfigurieren. Weitere Informationen finden Sie im [Amazon VPC-Benutzerhandbuch](#).

VPC bietet folgende Funktionen:

- Verarbeitung sensibler Daten

Das Starten eines Clusters in einer VPC ähnelt dem Starten des Clusters in einem privaten Netzwerk mit zusätzlichen Tools wie Routing-Tabellen und NetzwerkACLs, um zu definieren, wer Zugriff auf das Netzwerk hat. Wenn Sie sensible Daten in Ihrem Cluster verarbeiten, ist die zusätzliche Zugriffskontrolle, die das Starten von Ihrem Cluster in einer VPC bietet, nützlich. Außerdem können Sie Ihre Ressourcen in einem privaten Subnetz starten, in dem keine dieser Ressourcen über eine direkte Internetverbindung verfügt.

- Zugreifen auf Ressourcen in einem internen Netzwerk

Wenn sich Ihre Datenquelle in einem privaten Netzwerk befindet, kann es unpraktisch oder unerwünscht sein, diese Daten AWS für den Import in Amazon EMR hochzuladen, entweder aufgrund der zu übertragenden Datenmenge oder wegen der Vertraulichkeit der Daten. Stattdessen können Sie den Cluster in einer VPC starten und Ihr Rechenzentrum über eine VPN-Verbindung mit Ihrer VPC zu verbinden, damit der Cluster Zugriff auf Ressourcen im internen Netzwerk erhält. Wenn sich beispielsweise eine Oracle-Datenbank in Ihrem Rechenzentrum befindet, kann der Cluster auf diese Datenbank zugreifen, wenn Sie ihn in einer VPC starten, die mit dem entsprechenden Netzwerk über VPN verbunden ist.

Öffentliche und private Subnetze

Sie können Amazon-EMR-Cluster sowohl in öffentlichen als auch privaten VPC-Subnetzen starten. Das bedeutet, dass Sie keine Internetverbindung benötigen, um einen Amazon EMR-Cluster auszuführen. Möglicherweise müssen Sie jedoch Network Address Translation (NAT) und VPN-Gateways konfigurieren, um auf Dienste oder Ressourcen zuzugreifen, die sich außerhalb der VPC befinden, z. B. in einem Unternehmensintranet oder an Endpunkten für öffentliche AWS Dienste wie AWS Key Management Service

Important

Amazon EMR unterstützt nur das Starten von Clustern in privaten Subnetzen in den Versionen 4.2 oder höher.

Weitere Informationen zur Amazon VPC-Sicherheit finden Sie unter Sicherheit im [Amazon VPC-Benutzerhandbuch](#).

Themen

- [Amazon VPC-Optionen beim Starten eines Clusters](#)
- [Richten Sie eine VPC zum Hosten von Amazon EMR-Clustern ein](#)
- [Starten Sie Cluster in einer VPC mit Amazon EMR](#)
- [Beispielrichtlinien für private Subnetze, die auf Amazon S3 zugreifen](#)
- [Weitere Ressourcen, um mehr über Folgendes zu erfahren VPCs](#)

Amazon VPC-Optionen beim Starten eines Clusters

Wenn Sie ein Amazon-EMR-Cluster in einer VPC starten, können Sie diesen Vorgang in einem öffentlichen, privaten oder gemeinsamen Subnetz ausführen. Es gibt geringe, aber erwähnenswerte Unterschiede in Bezug auf die Konfiguration, je nachdem, welchen Subnetztyp Sie für ein Cluster auswählen.

Öffentliche Subnetze

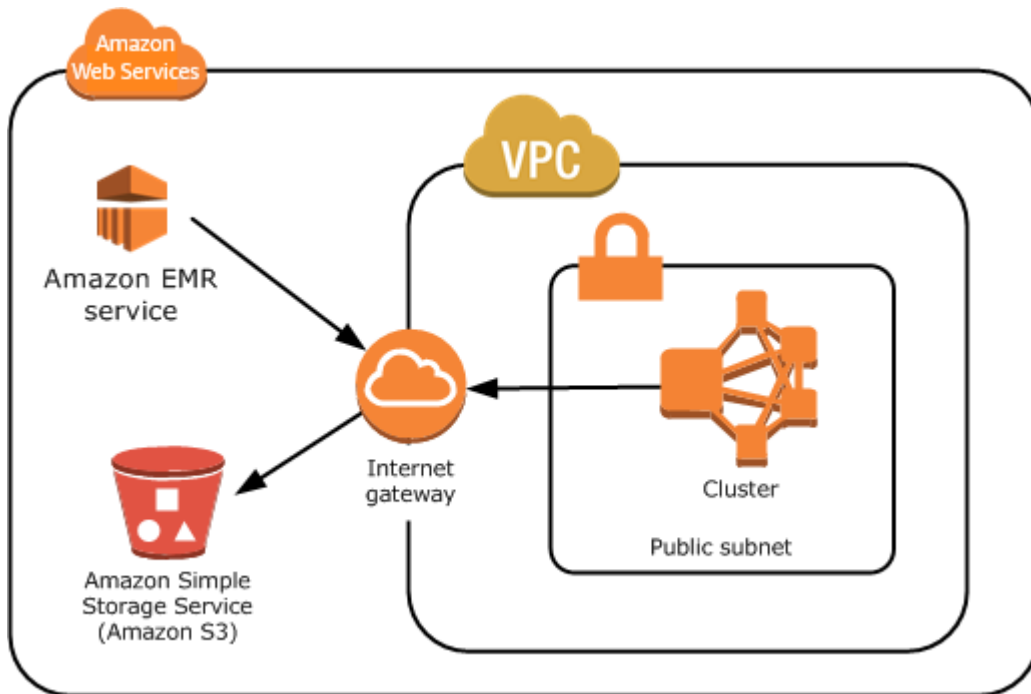
EMR-Cluster in einem öffentlichen Subnetz erfordern ein verbundenes Internet-Gateway. Dies liegt daran, dass Amazon EMR-Cluster auf AWS Services und Amazon EMR zugreifen müssen. Wenn ein Service, wie Amazon S3 die Möglichkeit zum Erstellen eines VPC-Endpunkts bietet, können Sie auf diese Services über den Endpunkt statt über einen öffentlichen Endpunkt und ein Internet-Gateway zugreifen. Darüber hinaus kann Amazon EMR nicht über ein NAT-Gerät (Network Address Translation) mit Clustern in öffentlichen Subnetzen kommunizieren. Zu diesem Zweck ist ein Internet-Gateway erforderlich. Sie können dennoch eine NAT-Instance oder ein Gateway für anderen Datenverkehr in komplexeren Szenarien verwenden.

Alle Instances in einem Cluster stellen entweder über einen VPC-Endpunkt oder ein Internet-Gateway eine Verbindung mit Amazon S3 her. Andere AWS Dienste, die derzeit keine VPC-Endpunkte unterstützen, verwenden nur ein Internet-Gateway.

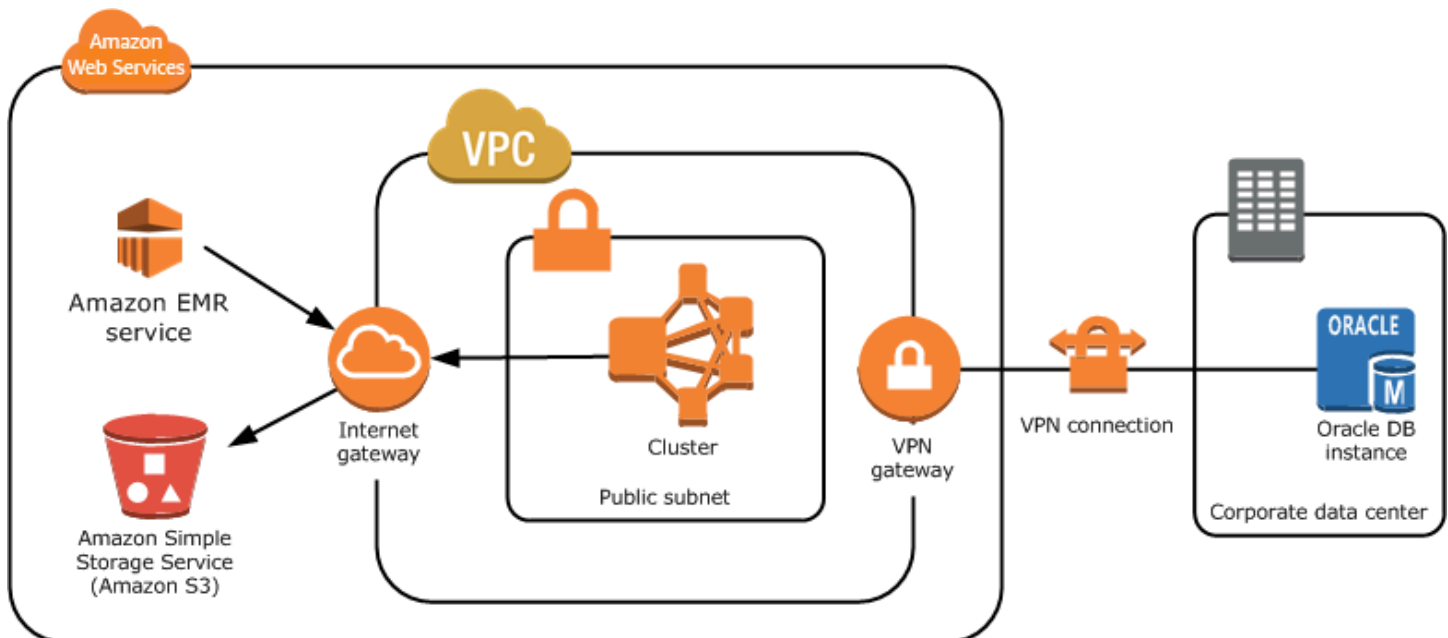
Wenn Sie über zusätzliche AWS Ressourcen verfügen, die nicht mit dem Internet-Gateway verbunden werden sollen, können Sie diese Komponenten in einem privaten Subnetz starten, das Sie in Ihrer VPC erstellen.

Cluster in einem öffentlichen Subnetz verwenden zwei Sicherheitsgruppen: eine für den Primärknoten und eine für Core- und Aufgabenknoten. Weitere Informationen finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Das folgende Diagramm zeigt, wie ein Amazon-EMR-Cluster in einer VPC mit einem öffentlichen Subnetz ausgeführt wird. Der Cluster kann über das Internet-Gateway eine Verbindung zu anderen AWS Ressourcen wie Amazon S3 S3-Buckets herstellen.



Das folgende Diagramm zeigt, wie Sie eine VPC so einrichten, dass ein Cluster in der VPC Zugriff auf Ressourcen in Ihrem eigenen Netzwerk, wie z. B. eine Oracle-Datenbank, hat.



Private Subnetze

Mit einem privaten Subnetz können Sie AWS Ressourcen starten, ohne dass das Subnetz über ein angeschlossenes Internet-Gateway verfügen muss. Amazon EMR unterstützt nur das Starten von Clustern in privaten Subnetzen in den Versionen 4.2.0 oder höher.

Note

Wenn Sie einen Amazon-EMR-Cluster in einem privaten Subnetz einrichten, empfehlen wir, dass Sie auch [VPC-Endpunkte für Amazon S3](#) einrichten. Wenn sich Ihr EMR-Cluster in einem privaten Subnetz ohne VPC-Endpunkte für Amazon S3 befindet, fallen zusätzliche NAT-Gateway-Gebühren an, die mit S3-Verkehr verbunden sind, da der Verkehr zwischen Ihrem EMR-Cluster und S3 nicht innerhalb Ihrer VPC verbleibt.

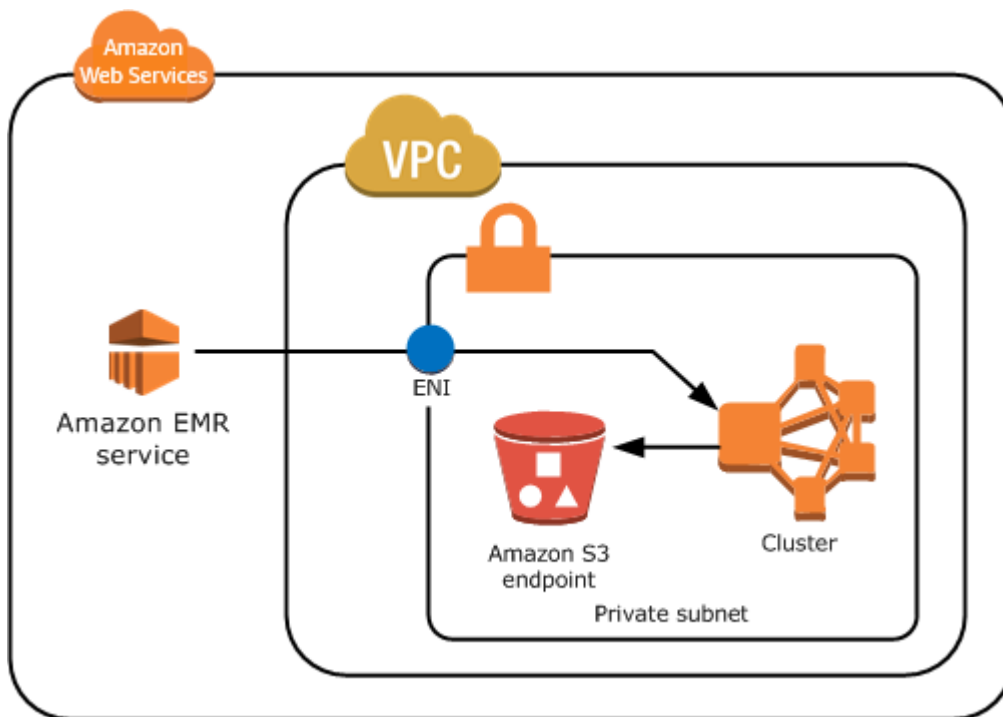
Private Subnetze unterscheiden sich von öffentlichen Subnetzen in folgenden Punkten:

- Um auf AWS Dienste zuzugreifen, die keinen VPC-Endpunkt bereitstellen, müssen Sie dennoch eine NAT-Instance oder ein Internet-Gateway verwenden.
- Sie müssen mindestens eine Route zum Amazon-EMR-Bucket für Serviceprotokolle und Amazon-Linux-Repository in Amazon S3 zur Verfügung stellen. Weitere Informationen finden Sie unter [Beispielrichtlinien für private Subnetze, die auf Amazon S3 zugreifen](#).
- Wenn Sie EMRFS-Features verwenden, benötigen Sie einen Amazon-S3-VPC-Endpunkt und eine Route von Ihrem privaten Subnetz zu DynamoDB.
- Debugging funktioniert nur, wenn Sie eine Route von Ihrem privaten Subnetz zu einem öffentlichen Amazon-SQS-Endpunkt bereitstellen.
- Das Erstellen eines privaten Subnetzes mit einer NAT-Instance oder einem Gateway in einem öffentlichen Subnetz wird nur mithilfe der AWS Management Console unterstützt. Der einfachste Weg, NAT-Instances und Amazon-S3-VPC-Endpunkte für Amazon-EMR-Cluster hinzuzufügen und zu konfigurieren, ist die Verwendung der Seite VPC-Subnetzliste in der Amazon-EMR-Konsole. Informationen zum Konfigurieren von [NAT-Gateways](#) finden Sie unter NAT-Gateways im Amazon-VPC-Benutzerhandbuch.
- Sie können ein Subnetz mit einem vorhandenen Amazon-EMR-Cluster nicht von öffentlich in privat oder umgekehrt ändern. Um ein Amazon-EMR-Cluster in einem privaten Subnetz zu finden, muss der Cluster in diesem privaten Subnetz gestartet werden.

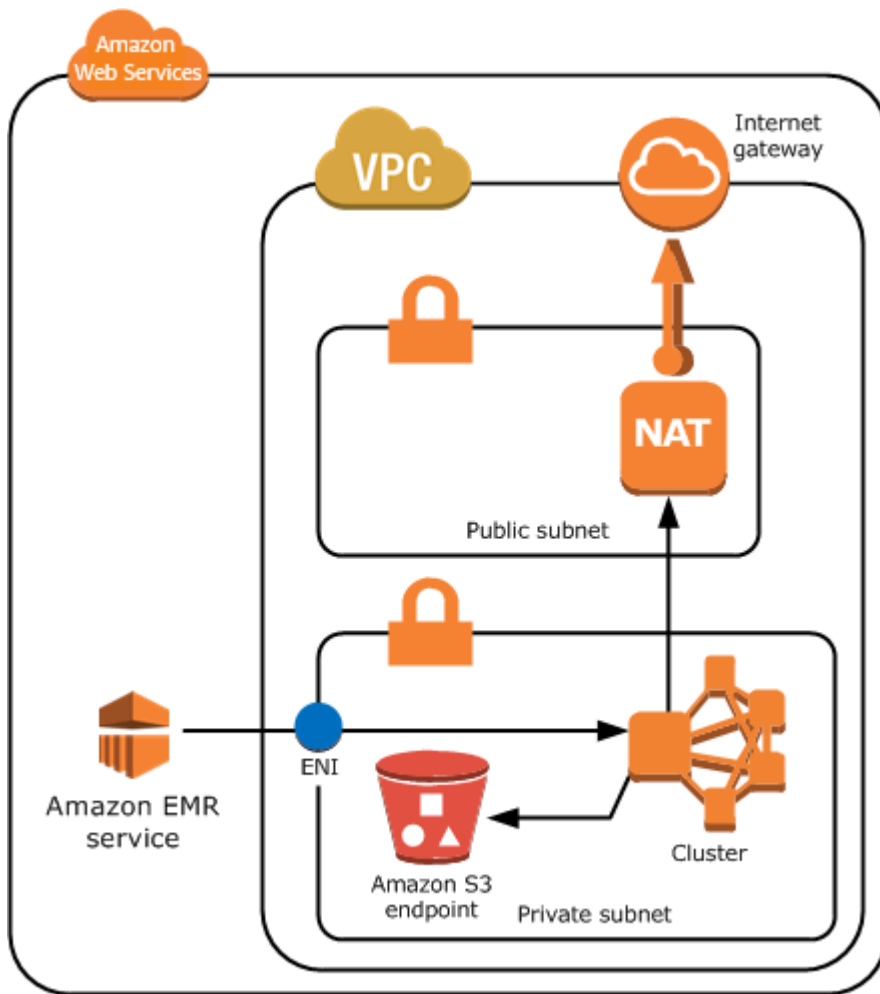
Amazon EMR erstellt und verwendet verschiedene Standardsicherheitsgruppen für die Cluster in einem privaten Subnetz: ElasticMapReduce-Master-Private, ElasticMapReduce-Slave-Private und -. ElasticMapReduce ServiceAccess Weitere Informationen finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Für eine vollständige Liste Ihres NACLs Clusters wählen Sie Sicherheitsgruppen für Primär und Sicherheitsgruppen für Core & Task auf der Seite Cluster-Details der Amazon EMR-Konsole.

Die folgende Abbildung zeigt, wie ein Amazon-EMR-Cluster in einem privaten Subnetz konfiguriert wird. Die einzige Kommunikation außerhalb des Subnetzes erfolgt mit Amazon EMR.



In der folgenden Abbildung ist eine Beispielkonfiguration für einen Amazon-EMR-Cluster in einem privaten Subnetz dargestellt, das mit einer NAT-Instance in einem öffentlichen Subnetz verbunden ist.



Gemeinsam genutzte Subnetze

VPC-Sharing ermöglicht es Kunden, Subnetze mit anderen AWS Konten innerhalb derselben AWS Organisation zu teilen. Sie können Amazon-EMR-Cluster sowohl in öffentlichen als auch privaten gemeinsamen Subnetzen starten, wobei folgende Einschränkungen gelten.

Der Subnetzbesitzer muss ein Subnetz für Sie freigeben, bevor Sie ein Amazon-EMR-Cluster darin starten können. Die Freigabe für gemeinsame Subnetze kann jedoch zu einem späteren Zeitpunkt wieder aufgehoben werden. Weitere Informationen finden Sie unter [Arbeiten mit Shared VPCs](#). Wenn ein Cluster in einem gemeinsamen Subnetz gestartet wird und die Freigabe für dieses gemeinsame Subnetz dann aufgehoben wird, lassen sich bei Aufhebung der Freigabe des Subnetzes je nach Status des Amazon-EMR-Clusters bestimmte Verhaltensweisen beobachten.

- Die Freigabe des Subnetzes wird vor dem erfolgreichen Start des Clusters aufgehoben – Wenn der Besitzer die Freigabe der Amazon VPC oder des Subnetzes aufhebt, während der Teilnehmer ein

Cluster startet, kann das Cluster nicht gestartet werden oder wird nur teilweise initialisiert, ohne alle angeforderten Instances bereitzustellen.

- Die Freigabe des Subnetzes wird nach dem erfolgreichen Start des Clusters aufgehoben – Wenn der Besitzer die Freigabe einer Amazon VPC oder eines Subnetzes für den Teilnehmer aufhebt, sind die Cluster des Teilnehmers nicht in der Lage, ihre Größe anzupassen, um neue Instances hinzuzufügen oder fehlerhafte Instances zu ersetzen.

Wenn Sie ein Amazon-EMR-Cluster starten, werden mehrere Sicherheitsgruppen erstellt. In einem gemeinsamen Subnetz steuert der Subnetzteilnehmer diese Sicherheitsgruppen. Der Subnetzbesitzer kann diese Sicherheitsgruppen zwar anzeigen, jedoch keine Aktionen bei diesen durchführen. Wenn der Subnetzbesitzer die Sicherheitsgruppe entfernen oder ändern möchte, muss der Teilnehmer, der die Sicherheitsgruppe erstellt hat, die Aktion durchführen.

VPC-Berechtigungen mit IAM steuern

Standardmäßig können alle -Benutzer sämtliche Subnetze für das Konto sehen einen Cluster in einem Subnetz starten.

Wenn Sie einen Cluster in einer VPC starten, können Sie AWS Identity and Access Management (IAM) verwenden, um den Zugriff auf Cluster zu kontrollieren und Aktionen mithilfe von Richtlinien einzuschränken, genau wie bei Clustern, die in Amazon EC2 Classic gestartet werden. Weitere Informationen zu IAM finden Sie im [IAM-Benutzerhandbuch](#).

Sie können auch IAM verwenden, um zu kontrollieren, wer Subnetze erstellen und verwalten kann. Sie können beispielsweise eine IAM-Rolle zur Verwaltung von Subnetzen und eine zweite Rolle erstellen, die Cluster starten, aber die Amazon VPC-Einstellungen nicht ändern kann. Weitere Informationen zur Verwaltung von Richtlinien und Aktionen in Amazon EC2 und Amazon VPC finden Sie unter [IAM-Richtlinien für Amazon EC2 im EC2 Amazon-Benutzerhandbuch](#).

Richten Sie eine VPC zum Hosten von Amazon EMR-Clustern ein

Bevor Sie Cluster in einer VPC starten können, müssen Sie eine VPC und ein Subnetz erstellen. Für öffentliche Subnetze müssen Sie ein Internet-Gateway erstellen und es dem Subnetz hinzufügen. Die folgenden Anweisungen beschreiben, wie Sie eine VPC erstellen, die Amazon-EMR-Cluster hosten können.

So erstellen Sie eine VPC mit Subnetzen für einen Amazon-EMR-Cluster

1. Öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.

2. Wählen Sie oben rechts auf der Seite [AWS-Region](#) für Ihre VPC aus.
3. Wählen Sie VPC erstellen aus.
4. Wählen Sie auf der Seite VPC-Einstellungen die Option VPC und mehr aus.
5. Aktivieren Sie unter Automatische Generierung von Namenstags die Option Automatisch generieren und geben Sie einen Namen für Ihre VPC ein. So können Sie die VPC und das Subnetz nach dem Erstellen in der Amazon VPC-Konsole identifizieren.
6. Geben Sie im Feld IPv4 CIDR-Block einen privaten IP-Adressraum für Ihre VPC ein, um eine korrekte DNS-Hostnamenauflösung sicherzustellen. Andernfalls kann es zu Ausfällen im Amazon EMR-Cluster kommen. Dieser Raum umfasst die folgenden IP-Adressbereiche:
 - 10.0.0.0 – 10.255.255.255
 - 172.16.0.0 – 172.31.255.255
 - 192.168.0.0 – 192.168.255.255
7. Wählen Sie unter Anzahl der Availability Zones (AZs) die Anzahl der Availability Zones aus, in denen Sie Ihre Subnetze starten möchten.
8. Wählen Sie unter Anzahl der öffentlichen Subnetze ein einzelnes öffentliches Subnetz aus, das Sie Ihrer VPC hinzufügen möchten. Wenn die vom Cluster verwendeten Daten im Internet verfügbar sind (z. B. in Amazon S3 oder Amazon RDS), müssen Sie nur ein öffentliches Subnetz verwenden und müssen kein privates Subnetz hinzufügen.
9. Wählen Sie unter Number of private subnets (Anzahl der privaten Subnetze) die Anzahl der privaten Subnetze aus, die Sie zu Ihrer VPC hinzufügen möchten. Wählen Sie eine oder mehrere aus, wenn die Daten für Ihre Anwendung in Ihrem eigenen Netzwerk gespeichert sind (z. B. in einer Oracle-Datenbank). Für eine VPC in einem privaten Subnetz müssen alle EC2 Amazon-Instances mindestens eine Route zu Amazon EMR über die elastic network interface haben. In der Konsole wird diese automatisch konfiguriert.
10. Wählen Sie unter NAT-Gateways optional, ob Sie NAT-Gateways hinzufügen möchten. Sie sind nur erforderlich, wenn Sie über private Subnetze verfügen, die mit dem Internet kommunizieren müssen.
11. Wählen Sie unter VPC-Endpunkte optional aus, Endpunkte für Amazon S3 zu Ihren Subnetzen hinzuzufügen.
12. Stellen Sie sicher, dass DNS-Hostnamen aktivieren und DNS-Auflösung aktivieren geprüft sind. Weitere Informationen finden Sie unter [Verwenden von DNS in Ihrer VPC](#).
13. Wählen Sie VPC erstellen aus.

14. Ein Statusfenster zeigt den Fortschritt an. Wenn die Arbeit abgeschlossen ist, wählen Sie View VPC aus, um zur VPCs Seite Ihre zu navigieren, auf der Ihre Standard-VPC und die VPC, die Sie gerade erstellt haben, angezeigt werden. Die zuvor erstellte VPC ist eine nicht standardmäßige VPC, daher wird in der Spalte Default VPC No angezeigt.
15. Wenn Sie Ihre VPC einem DNS-Eintrag zuordnen möchten, der keinen Domainnamen enthält, navigieren Sie zu DHCP-Optionssätzen, wählen Sie DHCP-Optionssatz erstellen aus und lassen Sie einen Domainnamen weg. Nachdem Sie Ihren Optionssatz erstellt haben, navigieren Sie zu Ihrer neuen VPC, wählen Sie im Menü Aktionen die Option DHCP-Optionssatz bearbeiten und wählen Sie den neuen Optionssatz aus. Sie können den Domainnamen nicht mit der Konsole bearbeiten, nachdem die DNS-Optionsliste festgelegt wurde.

Es ist eine bewährte Methode mit Hadoop und verwandten Anwendungen, um die Auflösung des Fully Qualified Domain Name (FQDN, vollständig qualifizierter Domainname) für Knoten sicherzustellen. Um die ordnungsgemäße DNS-Auflösung sicherzustellen, müssen Sie eine VPC konfigurieren, die eine DHCP-Optionsliste enthält, deren Parameter auf die folgenden Werte festgelegt sind:

- domain-name = **ec2.internal**

Verwenden Sie **ec2.internal**, wenn Ihre Region USA Ost (Nord-Virginia) ist.

Verwenden Sie für andere Regionen. *region-name* .**compute.internal** Beispiele für us-west-2 finden Sie in **us-west-2.compute.internal**. Verwenden **us-gov-west-1.compute.internal** Sie für die Region AWS GovCloud (US-West).

- domain-name-servers = **AmazonProvidedDNS**

Weitere Informationen finden Sie unter [DHCP-Optionssätze](#) im Amazon-VPC-Benutzerhandbuch.

16. Nachdem die VPC erstellt wurde, gehen Sie zur Seite Subnetze und notieren Sie die Subnetz-ID eines der Subnetze Ihrer neuen VPC. Diese Information benötigen Sie, wenn Sie den Amazon-EMR-Cluster in der VPC starten.

Starten Sie Cluster in einer VPC mit Amazon EMR

Nachdem Sie ein Subnetz zum Hosten von Amazon-EMR-Clustern konfiguriert haben, starten Sie den Cluster in diesem Subnetz, indem Sie die zugewiesene Subnetz-ID beim Erstellen des Clusters angeben.

 Note

Amazon EMR unterstützt private Subnetze in Version 4.2 und höher.

Wenn der Cluster gestartet wird, fügt Amazon EMR Sicherheitsgruppen hinzu, je nachdem, ob der Cluster in privaten oder öffentlichen Subnetzen der VPC gestartet wird. Alle Sicherheitsgruppen ermöglichen einen Zugang über Port 8443 für die Kommunikation mit dem Amazon-EMR-Service. Die IP-Adressbereiche sind jedoch für öffentliche und private Subnetze unterschiedlich. Amazon EMR verwaltet all diese Sicherheitsgruppen und muss dem AWS Bereich im Laufe der Zeit möglicherweise weitere IP-Adressen hinzufügen. Weitere Informationen finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Zum Verwalten des Cluster in einer VPC fügt Amazon EMR dem Primärknoten ein Netzwerkgerät an und verwaltet ihn über dieses Gerät. Sie können dieses Gerät mithilfe der Amazon EC2 API-Aktion anzeigen [DescribeInstances](#). Wenn Sie dieses Gerät ändern, fällt der Cluster möglicherweise aus.

Console

Um einen Cluster mit der Konsole in einer VPC zu starten

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Gehen Sie unter Networking zum Feld Virtual Private Cloud (VPC). Geben Sie den Namen Ihrer VPC ein oder wählen Sie Durchsuchen, um Ihre VPC auszuwählen. Wählen Sie alternativ VPC erstellen, um eine VPC zu erstellen, die Sie für Ihren Cluster verwenden können.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um einen Cluster in einer VPC zu starten mit dem AWS CLI

Note

Das AWS CLI bietet keine Möglichkeit, automatisch eine NAT-Instance zu erstellen und sie mit Ihrem privaten Subnetz zu verbinden. Um jedoch einen S3-Endpunkt in Ihrem Subnetz zu erstellen, können Sie die Amazon-VPC-CLI-Befehle nutzen. Verwenden Sie die Konsole zum Erstellen von NAT-Instances und Starten von Clustern in einem privaten Subnetz.

Nachdem Sie Ihre VPC konfiguriert haben, können Sie darin vorhandene Amazon-EMR-Cluster mithilfe des Unterbefehls `create-cluster` mit dem Parameter `--ec2-attributes` starten. Verwenden Sie den Parameter `--ec2-attributes`, um das VPC-Subnetz für den Cluster anzugeben.

- Um einen Cluster in einem bestimmten Subnetz zu erstellen, geben Sie den folgenden Befehl ein, *myKey* ersetzen Sie ihn durch den Namen Ihres EC2 Amazon-Schlüsselpaars und *77XXXX03* ersetzen Sie ihn durch Ihre Subnetz-ID.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --  
applications Name=Hadoop Name=Hive Name=Pig --use-default-roles --ec2-attributes  
KeyName=myKey,SubnetId=subnet-77XXXX03 --instance-type m5.xlarge --instance-  
count 3
```

Wenn Sie die Instance-Anzahl ohne den `--instance-groups`-Parameter angeben, wird ein einzelner Primärknoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

Note

Wenn Sie noch nicht die standardmäßige Amazon EMR-Servicerolle und das EC2 Instanzprofil erstellt haben, geben Sie ein, um sie `aws emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl eingeben.

Sicherstellung verfügbarer IP-Adressen für einen EMR-Cluster auf EC2

Um sicherzustellen, dass beim Start ein Subnetz mit ausreichend freien IP-Adressen verfügbar ist, überprüft die EC2 Subnetzauswahl die IP-Verfügbarkeit. Der Erstellungsprozess verwendet ein Subnetz mit der erforderlichen Anzahl an IP-Adressen, um Kern-, Primär- und Taskknoten nach Bedarf zu starten, auch wenn bei der ersten Erstellung nur Kernknoten für den Cluster erstellt werden. EMR überprüft während der Erstellung die Anzahl der IP-Adressen, die zum Starten von Primär- und Taskknoten erforderlich sind, und berechnet separat die Anzahl der IP-Adressen, die zum Starten von Kernknoten benötigt werden. Die Mindestanzahl der erforderlichen Primär- und Task-Instances oder Knoten wird automatisch von Amazon EMR bestimmt.

Important

Wenn keine Subnetze in der VPC ausreichend für wichtige Knoten verfügbar sind, wird ein Fehler zurückgegeben und der Cluster wird nicht erstellt.

In den meisten Bereitstellungsfällen gibt es einen Zeitunterschied zwischen den einzelnen Starts von Kern-, Primär- und Taskknoten. Darüber hinaus ist es möglich, dass sich mehrere Cluster ein Subnetz teilen. In diesen Fällen kann die Verfügbarkeit von IP-Adressen schwanken, sodass beispielsweise nachfolgende Task-Node-Starts durch verfügbare IP-Adressen eingeschränkt werden können.

Beispielrichtlinien für private Subnetze, die auf Amazon S3 zugreifen

Für private Subnetze müssen Sie Amazon EMR mindestens die Möglichkeit geben, auf Amazon-Linux-Repositorys zuzugreifen. Diese private Subnetzrichtlinie ist Teil der VPC-Endpunktrichtlinien für den Zugriff auf Amazon S3.

Mit Amazon EMR 5.25.0 oder höher müssen Sie Amazon EMR den Zugriff auf den System-Bucket ermöglichen, in dem die Spark-Ereignisprotokolle erfasst werden, um den Zugriff auf den persistenten Spark-History-Server mit nur einem Klick zu aktivieren. Wenn Sie die Protokollierung aktivieren, geben Sie PUT-Berechtigungen für den folgenden Bucket ein:

```
aws157-logs-${AWS::Region}/*
```

Weitere Informationen finden Sie unter [Zugriff auf den persistenten Spark History Server mit nur einem Klick](#).

Es ist dem Benutzer überlassen, den Businessanforderungen entsprechende Richtlinieneinschränkungen festzulegen. Die folgende Beispielrichtlinie bietet Berechtigungen für den Zugriff auf Amazon-Linux-Repositorys und den Amazon-EMR-System-Bucket zum Sammeln von Spark-Ereignisprotokollen. Es werden einige Beispiele für Ressourcennamen für die Buckets angezeigt.

Weitere Informationen zur Verwendung von IAM-Richtlinien mit Amazon-VPC-Endpunkten finden Sie unter [Endpunktrichtlinien für Amazon S3](#).

Das folgende Richtlinienbeispiel enthält Beispielressourcen in der Region us-east-1.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AmazonLinuxAMIRepositoryAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::packages.us-east-1.amazonaws.com/*",
        "arn:aws:s3:::repo.us-east-1.amazonaws.com/",
        "arn:aws:s3:::repo.us-east-1.amazonaws.com/*"
      ]
    },
    {
      "Sid": "EnableApplicationHistory",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:Put*",
        "s3:Get*",
        "s3:Create*",
        "s3:Abort*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::prod.us-east-1.appinfo.src/*"
      ]
    }
  ]
}
```

Die folgende Beispielrichtlinie stellt die Berechtigungen bereit, die für den Zugriff auf Amazon-Linux-2-Repositorys erforderlich sind. Amazon-Linux-2-AMI ist die Standardeinstellung.

```
{
  "Statement": [
    {
      "Sid": "AmazonLinux2AMIRepositoryAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": [
        "arn:aws:s3:::amazonlinux.us-east-1.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux-2-repos-us-east-1/*"
      ]
    }
  ]
}
```

Verfügbare Regionen

Die folgende Tabelle enthält eine Liste von Buckets nach Regionen und enthält sowohl einen Amazon-Ressourcennamen (ARN) für das Repository als auch eine Zeichenfolge, die den ARN für den darstellt. appinfo.src Der ARN oder Amazon Resource Name ist eine Zeichenfolge, die eine AWS Ressource eindeutig identifiziert.

Region	Repository-Buckets	AppInfo Eimer
USA Ost (Ohio)	„arn:aws:s3::: packages.us-east-2 .amazonaws.com/“, "arn:aws:s3::: repo.us-east-2.amazonaws.com/“, "arn:aws:s3: ::repo.us-east-2.e mr.amazonaws.com/*“	„arn:aws:s3: ::prod.us-east-2.a ppinfo.src/*“
USA Ost (Nord-Virginia)	„arn:aws:s3::: packages.us-east-1 .amazonaws.com/“, "arn:aws:s3::: repo.us-east-1.amazonaws.com/“, "arn:aws:s3: ::repo.us-east-1.e mr.amazonaws.com/*“	„arn:aws:s3: ::prod.us-east-1.a ppinfo.src/*“

Region	Repository-Buckets	AppInfo Eimer
USA West (Nordkalifornien)	„arn:aws:s3::: packages.us-west-1.amazonaws.com/“, "arn:aws:s3::: repo.us-west-1.amazonaws.com/“, "arn:aws:s3: ::repo.us-west-1.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.us-west-1.appinfo.src/*"
USA West (Oregon)	„arn:aws:s3::: packages.us-west-2.amazonaws.com/“, "arn:aws:s3::: repo.us-west-2.amazonaws.com/“, "arn:aws:s3: ::repo.us-west-2.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.us-west-2.appinfo.src/*"
Afrika (Kapstadt)	„arn:aws:s3::: packages.af-south-1.amazonaws.com/“, "arn:aws:s3::: repo.af-south-1.amazonaws.com/“, "arn:aws:s3: ::repo.af-south-1.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.af-south-1.appinfo.src/*"
Afrika (Kapstadt)	„arn:aws:s3::: packages.ap-east-1.amazonaws.com/“, "arn:aws:s3::: repo.ap-east-1.amazonaws.com/“, "arn:aws:s3: ::repo.ap-east-1.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.ap-east-1.appinfo.src/*"
Asien-Pazifik (Hyderabad)	„arn:aws:s3::: packages.ap-south-2.amazonaws.com/“, "arn:aws:s3::: repo.ap-south-2.amazonaws.com/“, "arn:aws:s3: ::repo.ap-south-2.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.ap-south-2.appinfo.src/*"
Asien-Pazifik (Jakarta)	„arn:aws:s3::: packages.ap-southeast-3.amazonaws.com/“, "arn:aws:s3::: repo.ap-southeast-3.amazonaws.com/“, "arn:aws:s3: ::repo.ap-southeast-3.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.ap-southeast-3.appinfo.src/*"

Region	Repository-Buckets	AppInfo Eimer
Asien-Pazifik (Malaysia)	„arn:aws:s3::: packages.ap-southeast-5.amazonaws.com/“, "arn:aws:s3::: repo.ap-southeast-5.amazonaws.com/“, "arn:aws:s3:::repo.ap-southeast-5.emr.amazonaws.com/*"	„arn:aws:s3:::prod.ap-southeast-5.appinfo.src/*"
Asien-Pazifik (Melbourne)	„arn:aws:s3::: packages.ap-southeast-4.amazonaws.com/“, "arn:aws:s3::: repo.ap-southeast-4.amazonaws.com/“, "arn:aws:s3:::repo.ap-southeast-4.emr.amazonaws.com/*"	„arn:aws:s3:::prod.ap-south-4.appinfo.src/*"
Asien-Pazifik (Mumbai)	„arn:aws:s3::: packages.ap-south-1.amazonaws.com/“, "arn:aws:s3::: repo.ap-south-1.amazonaws.com/“, "arn:aws:s3:::repo.ap-south-1.emr.amazonaws.com/*"	„arn:aws:s3:::prod.ap-south-1.appinfo.src/*"
Asien-Pazifik (Osaka)	„arn:aws:s3::: packages.ap-northeast-3.amazonaws.com/“, "arn:aws:s3::: repo.ap-northeast-3.amazonaws.com/“, "arn:aws:s3:::repo.ap-northeast-3.emr.amazonaws.com/*"	„arn:aws:s3:::prod.ap-northeast-3.appinfo.src/*"
Asien-Pazifik (Seoul)	„arn:aws:s3::: packages.ap-northeast-2.amazonaws.com/“, "arn:aws:s3::: repo.ap-northeast-2.amazonaws.com/“, "arn:aws:s3:::repo.ap-northeast-2.emr.amazonaws.com/*"	„arn:aws:s3:::prod.ap-northeast-2.appinfo.src/*"
Asien-Pazifik (Singapur)	„arn:aws:s3::: packages.ap-southeast-1.amazonaws.com/“, "arn:aws:s3::: repo.ap-southeast-1.amazonaws.com/“, "arn:aws:s3:::repo.ap-southeast-1.emr.amazonaws.com/*"	„arn:aws:s3:::prod.ap-southeast-1.appinfo.src/*"

Region	Repository-Buckets	AppInfo Eimer
Asien-Pazifik (Sydney)	„arn:aws:s3::: packages.ap-southeast-2.amazonaws.com/“, "arn:aws:s3::: repo.ap-southeast-2.amazonaws.com/“, "arn:aws:s3:::repo.ap-southeast-2.emr.amazonaws.com/*“	„arn:aws:s3:::prod.ap-southeast-2.appinfo.src/*“
Asien-Pazifik (Tokio)	„arn:aws:s3::: packages.ap-northeast-1.amazonaws.com/“, "arn:aws:s3::: repo.ap-northeast-1.amazonaws.com/“, "arn:aws:s3:::repo.ap-northeast-1.emr.amazonaws.com/*“	„arn:aws:s3:::prod.ap-northeast-1.appinfo.src/*“
Kanada (Zentral)	„arn:aws:s3::: packages.ca-central-1.amazonaws.com/“, "arn:aws:s3::: repo.ca-central-1.amazonaws.com/“, "arn:aws:s3:::repo.ca-central-1.emr.amazonaws.com/*“	„arn:aws:s3:::prod.ca-central-1.appinfo.src/*“
Kanada West (Calgary)	„arn:aws:s3::: packages.ca-west-1.amazonaws.com/“, "arn:aws:s3::: repo.ca-west-1.amazonaws.com/“, "arn:aws:s3:::repo.ca-west-1.emr.amazonaws.com/*“	„arn:aws:s3:::prod.ca-west-1.appinfo.src/*“
Europa (Frankfurt)	„arn:aws:s3::: packages.eu-central-1.amazonaws.com/“, "arn:aws:s3::: repo.eu-central-1.amazonaws.com/“, "arn:aws:s3:::repo.eu-central-1.emr.amazonaws.com/*“	„arn:aws:s3:::prod.eu-central-1.appinfo.src/*“
Europa (Irland)	„arn:aws:s3::: packages.eu-west-1.amazonaws.com/“, "arn:aws:s3::: repo.eu-west-1.amazonaws.com/“, "arn:aws:s3:::repo.eu-west-1.emr.amazonaws.com/*“	„arn:aws:s3:::prod.eu-west-1.appinfo.src/*“

Region	Repository-Buckets	AppInfo Eimer
Europa (London)	„arn:aws:s3::: packages.eu-west-2.amazonaws.com/“, "arn:aws:s3::: repo.eu-west-2.amazonaws.com/“, "arn:aws:s3: ::repo.eu-west-2.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.eu-west-2.appinfo.src/*"
Europa (Milan)	„arn:aws:s3::: packages.eu-south-1.amazonaws.com/“, "arn:aws:s3::: repo.eu-south-1.amazonaws.com/“, "arn:aws:s3: ::repo.eu-south-1.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.eu-south-1.appinfo.src/*"
Europa (Paris)	„arn:aws:s3::: packages.eu-west-3.amazonaws.com/“, "arn:aws:s3::: repo.eu-west-3.amazonaws.com/“, "arn:aws:s3: ::repo.eu-west-3.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.eu-west-3.appinfo.src/*"
Europa (Spain)	„arn:aws:s3::: packages.eu-south-2.amazonaws.com/“, "arn:aws:s3::: repo.eu-south-2.amazonaws.com/“, "arn:aws:s3: ::repo.eu-south-2.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.eu-south-2.appinfo.src/*"
Europa (Stockholm)	„arn:aws:s3::: packages.eu-north-1.amazonaws.com/“, "arn:aws:s3::: repo.eu-north-1.amazonaws.com/“, "arn:aws:s3: ::repo.eu-north-1.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.eu-north-1.appinfo.src/*"
Europa (Zürich)	„arn:aws:s3::: packages.eu-central-2.amazonaws.com/“, "arn:aws:s3::: repo.eu-central-2.amazonaws.com/“, "arn:aws:s3: ::repo.eu-central-2.emr.amazonaws.com/*"	„arn:aws:s3: ::prod.eu-central-2.appinfo.src/*"

Region	Repository-Buckets	AppInfo Eimer
Israel (Tel Aviv)	„arn:aws:s3::: packages.il-central-1.amazonaws.com/“, "arn:aws:s3::: repo.il-central-1.amazonaws.com/“, "arn:aws:s3: ::repo.il-central-1.emr.amazonaws.com/*“	„arn:aws:s3: ::prod.il-central-1.appinfo.src/*“
Naher Osten (Bahrain)	„arn:aws:s3::: packages.me-south-1.amazonaws.com/“, "arn:aws:s3::: repo.me-south-1.amazonaws.com/“, "arn:aws:s3: ::repo.me-south-1.emr.amazonaws.com/*“	„arn:aws:s3: ::prod.me-south-1.appinfo.src/*“
Naher Osten (VAE)	„arn:aws:s3::: packages.me-central-1.amazonaws.com/“, "arn:aws:s3::: repo.me-central-1.amazonaws.com/“, "arn:aws:s3: ::repo.me-central-1.emr.amazonaws.com/*“	„arn:aws:s3: ::prod.me-central-1.appinfo.src/*“
Südamerika (São Paulo)	„arn:aws:s3::: packages.sa-east-1.amazonaws.com/“, "arn:aws:s3::: repo.sa-east-1.amazonaws.com/“, "arn:aws:s3: ::repo.sa-east-1.emr.amazonaws.com/*“	„arn:aws:s3: ::prod.sa-east-1.appinfo.src/*“
AWS GovCloud (US-Ost)	„arn:aws:s3: ::Pakete. us-gov-east-1.amazonaws.com/“, "arn:aws:s3: ::repo. us-gov-east-1.amazonaws.com/“, "arn:aws:s3: ::repo. us-gov-east-1.emr.amazonaws.com/*“	„arn:aws:s3: ::prod. us-gov-east-1.appinfo.src/*“
AWS GovCloud (US-West)	„arn:aws:s3: ::Pakete. us-gov-west-1.amazonaws.com/“, "arn:aws:s3: ::repo. us-gov-west-1.amazonaws.com/“, "arn:aws:s3: ::repo. us-gov-west-1.emr.amazonaws.com/*“	„arn:aws:s3: ::prod.me-south-1.appinfo.src/*“

Weitere Ressourcen, um mehr über Folgendes zu erfahren VPCs

In den folgenden Themen erfahren Sie mehr über VPCs Subnetze.

- Private Subnetze in einer VPC
 - [Szenario 2: VPC mit öffentlichen und privaten Subnetzen \(NAT\)](#)
 - [NAT-Instances](#)
 - [Hohe Verfügbarkeit für Amazon VPC NAT-Instances: Beispiel](#)
- Öffentliche Subnetze in einer VPC
 - [Szenario 1: VPC mit einem einzelnen öffentlichen Subnetz](#)
- Allgemeine VPC-Informationen
 - [Amazon VPC User Guide](#)
 - [VPC Peering](#)
 - [Verwenden von Elastic Network-Schnittstellen mit Ihrer VPC](#)
 - [Sichere Verbindung mit Linux-Instances, die in einer privaten VPC ausgeführt werden](#)

Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen

Wenn Sie einen Cluster erstellen und die Konfiguration der Primär-, Core- und Aufgabenknoten angeben, stehen Ihnen zwei Konfigurationsoptionen zur Verfügung. Sie können Instance-Flotten oder einheitliche Instance-Gruppen verwenden. Die Konfigurationsoption, die Sie auswählen, gilt für alle Knoten und für die gesamte Nutzungsdauer des Clusters. Instance-Flotten und Instance-Gruppen können nicht gleichzeitig in einem Cluster vorhanden sein. Die Instance-Flottenkonfiguration ist in Amazon-EMR-Versionen 4.8.0 und höher verfügbar, nicht in 5.0.x-Versionen.

Sie können die Amazon EMR-Konsole, die oder die Amazon EMR-API verwenden AWS CLI, um Cluster mit einer der beiden Konfigurationen zu erstellen. Wenn Sie den Befehl `create-cluster` in der AWS CLI verwenden, erstellen Sie den Cluster mit den `--instance-fleets`-Parametern, um den Cluster mittels Instance-Flotten zu erstellen, oder mit den `--instance-groups`-Parametern, um den Cluster mittels einheitlicher Instance-Gruppen zu erstellen.

Dasselbe gilt auch, wenn Sie die Amazon-EMR-API verwenden. Sie geben mit der Konfiguration `InstanceGroups` eine Reihe von `InstanceGroupConfig`-Objekten oder mit der Konfiguration `InstanceFleets` eine Reihe von `InstanceFleetConfig`-Objekten an.

In der neuen Amazon-EMR-Konsole können Sie wählen, ob Sie bei der Erstellung eines Clusters entweder Instance-Gruppen oder Instance-Flotten verwenden möchten, und Sie haben die Möglichkeit, Spot Instances mit jeder zu verwenden. Wenn Sie mit der alten Amazon-EMR-Konsole beim Erstellen Ihres Clusters die Standardeinstellungen für Schnelloptionen verwenden, wendet Amazon EMR die einheitliche Instance-Gruppenkonfiguration auf den Cluster an und verwendet On-Demand-Instances. Um Spot-Instances mit einheitlichen Instance-Gruppen zu verwenden oder Instance-Flotten und anderen Anpassungen zu konfigurieren, wählen Sie Advanced Options (Erweiterte Optionen) aus.

Instance-Flotten

Die Konfiguration der Instance-Flotten bietet die größte Vielfalt an Bereitstellungsoptionen für Amazon-Instances. EC2 Jeder Knotentyp verfügt über eine einzelne Instance-Flotte und die Aufgaben-Instance-Flotte ist optional. Sie können bis zu fünf EC2 Instance-Typen pro Flotte oder 30 EC2 Instance-Typen pro Flotte angeben, wenn Sie einen Cluster mithilfe der AWS CLI oder Amazon EMR-API und einer [Zuweisungsstrategie](#) für On-Demand- und Spot-Instances erstellen. Für die Core- und Aufgaben-Instance-Flotten weisen Sie eine Zielkapazität für On-Demand-Instances und eine zweite für Spot Instances zu. Amazon EMR wählt eine Kombination aus fünf Instance-Typen aus, um die Zielkapazitäten zu erfüllen und stellt On-Demand- als auch Spot Instances bereit.

Für den Primärknoten-Typ wählt Amazon EMR einen einzelnen Instance-Typ aus Ihrer Liste von bis zu fünf Typen aus und Sie geben an, ob diese als On-Demand- oder Spot Instance bereitgestellt werden sollen. Instance-Flotten bieten auch zusätzliche Optionen für Spot Instance- und On-Demand-Käufe. Zu den Spot-Instance-Optionen gehören ein Timeout, das festlegt, welche Maßnahme ergriffen werden soll, wenn Spot-Kapazität nicht bereitgestellt werden kann, und eine bevorzugte Zuweisungsstrategie (kapazitätsoptimiert) für den Start von Spot Instance-Flotten. On-Demand-Instance-Flotten können auch mit der Option der Zuweisungsstrategie (niedrigster Preis) gestartet werden. Wenn Sie eine Servicerolle verwenden, die nicht die EMR-Standardservicerolle ist, oder eine von EMR verwaltete Richtlinie in Ihrer Servicerolle verwenden, müssen Sie der benutzerdefinierten Clusterservicerolle zusätzliche Berechtigungen hinzufügen, um die Option für die Zuweisungsstrategie zu aktivieren. Weitere Informationen finden Sie unter [Servicerolle für Amazon EMR \(EMR-Rolle\)](#).

Weitere Information zum Konfigurieren von Instance-Flotten finden Sie unter [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#).

Einheitliche Instance-Gruppen

Einheitliche Instance-Gruppen bieten eine einfachere Einrichtung als Instance-Flotten. Jeder Amazon EMR-Cluster kann bis zu 50 Instance-Gruppen umfassen: eine primäre Instance-Gruppe, die eine EC2 Amazon-Instance enthält, eine Core-Instance-Gruppe, die eine oder mehrere EC2 Instances enthält, und bis zu 48 optionale Task-Instance-Gruppen. Jede Kern- und Task-Instance-Gruppe kann eine beliebige Anzahl von EC2 Amazon-Instances enthalten. Sie können jede Instance-Gruppe skalieren, indem Sie EC2 Amazon-Instances manuell hinzufügen und entfernen, oder Sie können die automatische Skalierung einrichten. Weitere Informationen über das Hinzufügen und Entfernen von Instances finden Sie unter [Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen](#).

Weitere Informationen zum Konfigurieren von einheitlichen Instance-Gruppen finden Sie unter [Konfigurieren Sie einheitliche Instanzgruppen für Ihren Amazon EMR-Cluster](#).

Arbeiten mit Instance-Flotten und Instance-Gruppen

Themen

- [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#)
- [Neukonfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#)
- [Verwenden Sie Kapazitätsreservierungen mit Instance-Flotten in Amazon EMR](#)
- [Konfigurieren Sie einheitliche Instanzgruppen für Ihren Amazon EMR-Cluster](#)
- [Flexibilität der Availability Zone für einen Amazon EMR-Cluster](#)
- [Konfiguration von Amazon EMR-Cluster-Instance-Typen und Best Practices für Spot-Instances](#)

Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster

Note

Die Konfiguration der Instance-Flotten ist nur in den Amazon-EMR-Versionen 4.8.0 und höher verfügbar, mit Ausnahme von 5.0.0 und 5.0.3.

Mit der Instance-Flottenkonfiguration für Amazon EMR-Cluster können Sie eine Vielzahl von Bereitstellungsoptionen für EC2 Amazon-Instances auswählen und eine flexible und elastische Ressourcenstrategie für jeden Knotentyp in Ihrem Cluster entwickeln.

In einer Instance-Flottenkonfiguration geben Sie eine Zielkapazität für [On-Demand-Instances](#) und [Spot Instances](#) innerhalb jeder Flotte an. Wenn der Cluster gestartet wird, stellt Amazon EMR Instances bereit, bis die Ziele erfüllt sind. Wenn Amazon aufgrund einer Preiserhöhung oder eines Instance-Fehlers eine Spot-Instance in einem laufenden Cluster EC2 zurückfordert, versucht Amazon EMR, die Instance durch einen der von Ihnen angegebenen Instance-Typen zu ersetzen. Dies erleichtert die Wiedererlangung der Kapazität während eines Anstiegs der Spot-Preise.

Sie können maximal fünf EC2 Amazon-Instance-Typen pro Flotte angeben, die Amazon EMR bei der Erfüllung der Ziele verwenden soll, oder maximal 30 EC2 Amazon-Instance-Typen pro Flotte, wenn Sie einen Cluster mithilfe der AWS CLI oder Amazon EMR-API und einer [Zuweisungsstrategie](#) für On-Demand- und Spot-Instances erstellen.

Sie können auch mehrere Subnetze für verschiedene Availability Zones auswählen. Wenn Amazon EMR den Cluster startet, sieht er auf diese Subnetze, um die Instances und Kaufoptionen zu finden, die Sie angeben. Wenn Amazon EMR ein AWS großes Ereignis in einer oder mehreren Availability Zones erkennt, versucht Amazon EMR automatisch, den Verkehr von den betroffenen Availability Zones wegzuleiten und neue Cluster zu starten, die Sie entsprechend Ihrer Auswahl in alternativen Availability Zones erstellen. Beachten Sie, dass die Auswahl der Cluster-Availability-Zone nur bei der Cluster-Erstellung erfolgt. Bestehende Clusterknoten werden bei einem Ausfall der Availability Zone nicht automatisch in einer neuen Availability Zone neu gestartet.

Überlegungen zur Arbeit mit Instance-Flotten

Berücksichtigen Sie Folgendes, wenn Sie Instance-Flotten mit Amazon EMR verwenden.

- Sie können eine Instance-Flotte haben, und zwar nur eine pro Knotentyp (Primär, Core, Aufgabe). Sie können bis zu fünf EC2 Amazon-Instance-Typen für jede Flotte auf der angegebenen AWS Management Console (oder maximal 30 Typen pro Instance-Flotte, wenn Sie einen Cluster mithilfe der AWS CLI oder Amazon EMR-API und einer erstellen [Zuweisungsstrategie für Flotten](#)).
- Amazon EMR wählt einen oder alle der angegebenen EC2 Amazon-Instance-Typen aus, um sowohl Spot- als auch On-Demand-Kaufoptionen bereitzustellen.
- Legen Sie Zielkapazitäten für Spot- und On-Demand-Instances für die Core- und Aufgaben-Flotte fest. Verwenden Sie vCPU oder eine generische Einheit, die jeder EC2 Amazon-Instance zugewiesen ist, die auf die Ziele angerechnet wird. Amazon EMR stellt Instances bereit, bis die jeweilige Zielkapazität völlig erfüllt ist. Für die Primär-Flotte ist das Ziel immer auf 1 gesetzt.
- Sie können ein Subnetz (Availability Zone) oder einen Bereich auswählen. Amazon EMR stellt Kapazität in der Availability Zone bereit, die am besten passt.
- Hinweise zum Angeben der Zielkapazität für Spot-Instances:

- Bestimmen Sie für jeden Instance-Typ einen maximalen Spot-Preis. Amazon EMR stellt Spot Instances bereit, wenn der Spot-Preis unter dem maximalen Spot-Preis liegt. Sie zahlen den Spot-Preis, nicht unbedingt den maximalen Spot-Preis.
- Für jede Flotte definieren Sie einen Timeout-Zeitraum für die Bereitstellung von Spot-Instances. Wenn Amazon EMR keine Spot-Kapazität bereitstellen kann, können Sie den Cluster beenden oder stattdessen zur Bereitstellung von On-Demand-Kapazität wechseln. Dies gilt nur für die Bereitstellung von Clustern, nicht für deren Größenänderung. Wenn der Timeout-Zeitraum während der Größenänderung des Clusters endet, werden Spot-Anfragen, die nicht bereitgestellt wurden, für ungültig erklärt, ohne dass sie auf On-Demand-Kapazität übertragen werden.
- Für jede Flotte können Sie eine der folgenden Zuweisungsstrategien für Ihre Spot-Instances angeben: preiskapazitätsoptimiert, kapazitätsoptimiert capacity-optimized-prioritized, niedrigster Preis oder diversifiziert über alle Pools hinweg.
- Für jede Flotte können Sie die folgenden Zuweisungsstrategien für Ihre On-Demand-Instances anwenden: die Strategie mit dem niedrigsten Preis oder die Strategie mit Priorität.
- Für jede Flotte mit On-Demand-Instances können Sie wählen, ob Sie Optionen zur Kapazitätsreservierung anwenden möchten.
- Wenn Sie eine Zuweisungsstrategie für Instance-Flotten verwenden, müssen Sie bei der Auswahl von Subnetzen für Ihren EMR-Cluster die folgenden Überlegungen beachten:
 - Wenn Amazon EMR einen Cluster mit einer Task-Flotte bereitstellt, filtert es Subnetze heraus, denen genügend verfügbare IP-Adressen fehlen, um alle Instances des angeforderten EMR-Clusters bereitzustellen. Dazu gehören IP-Adressen, die für die Primär-, Kern- und Task-Instance-Flotten beim Cluster-Start erforderlich sind. Amazon EMR nutzt dann seine Zuweisungsstrategie, um den Instance-Pool auf der Grundlage des Instance-Typs und der verbleibenden Subnetze mit ausreichend IP-Adressen zu bestimmen, um den Cluster zu starten.
 - Wenn Amazon EMR aufgrund unzureichender verfügbarer IP-Adressen nicht den gesamten Cluster starten kann, versucht es, Subnetze mit ausreichend freien IP-Adressen zu identifizieren, um die wesentlichen Instance-Flotten (Kern- und Primär-) zu starten. In solchen Szenarien wird Ihre Task-Instance-Flotte in einen angehaltenen Zustand versetzt, anstatt den Cluster mit einem Fehler zu beenden.
 - Wenn keines der angegebenen Subnetze genügend IP-Adressen für die Bereitstellung der wesentlichen Core- und Primärinstanzflotten enthält, schlägt der Clusterstart mit einem `VALIDATION_ERROR` fehl. Dadurch wird ein Clusterabbruch mit dem Schweregrad KRITISCH ausgelöst und Sie werden darüber informiert, dass der Cluster nicht gestartet werden kann. Um dieses Problem zu vermeiden, empfehlen wir, die Anzahl der IP-Adressen in Ihren Subnetzen zu erhöhen.

- Wenn Sie die Amazon EMR-Version emr-7.7.0 und höher ausführen und eine Zuweisungsstrategie für Instance-Flotten verwenden, können Sie den Cluster auf bis zu 4000 EC2 Instances und 14000 EBS-Volumes pro Instance-Flotte skalieren. Für Release-Versionen unter EMR-7.7.0 kann der Cluster nur auf 2000 Instances und 7000 EBS-Volumes pro Instance-Flotte hochskaliert werden. EC2
- Wenn Sie On-Demand-Instances starten, können Sie offene oder gezielte Kapazitätsreservierungen für Primär-, Kern- und Task-Knoten in Ihren Konten verwenden. Bei On-Demand-Instances mit einer Zuweisungsstrategie für Instance-Flotten ist die Kapazität möglicherweise nicht ausreichend. Wir empfehlen, dass Sie mehrere Instance-Typen angeben, um zu diversifizieren und das Risiko einer unzureichenden Kapazität zu verringern. Weitere Informationen finden Sie unter [the section called “Verwenden Sie Kapazitätsreservierungen mit Instance-Flotten in Amazon EMR”](#).

Instance-Flotten-Optionen

Verwenden Sie die folgenden Richtlinien, um Instance-Flotten-Optionen zu verstehen.

Themen

- [Festlegen von Zielkapazitäten](#)
- [Start-Optionen](#)
- [Optionen für mehrere Subnetze \(Availability Zones\)](#)
- [Hauptknoten-Konfiguration](#)

Festlegen von Zielkapazitäten

Geben Sie die Zielkapazitäten für die Core- und Aufgaben-Flotte an. Wenn Sie dies tun, wird die Anzahl der On-Demand-Instances und Spot Instances festgelegt, die Amazon EMR bereitstellt. Wenn Sie eine Instance angeben, können Sie entscheiden, wie viel jede Instance beim Ziel mit eingerechnet wird. Wenn eine On-Demand-Instance bereitgestellt wird, wird sie beim On-Demand-Ziel mit eingerechnet. Dies gilt auch für Spot-Instances. Im Gegensatz zu Core- und Aufgaben-Flotten besteht die Primär-Flotte immer aus einer Instance. Daher ist die Zielkapazität für diese Flotte immer auf 1 gesetzt.

Wenn Sie die Konsole verwenden, wird standardmäßig das v CPUs des EC2 Amazon-Instance-Typs als Anzahl für die Zielkapazitäten verwendet. Sie können dies in Generische Einheiten ändern und dann die Anzahl für jeden EC2 Instance-Typ angeben. Wenn Sie den verwenden AWS CLI, weisen Sie jedem Instanztyp manuell generische Einheiten zu.

 Important

Wenn Sie mithilfe von einem Instance-Typ auswählen AWS Management Console, entspricht die Anzahl der vCPUs, die für jeden Instance-Typ angezeigt wird, der Anzahl der YARN-vcores für diesen Instance-Typ, nicht der Anzahl von EC2 v CPUs für diesen Instance-Typ. Weitere Informationen zur Anzahl von v CPUs für jeden Instance-Typ finden Sie unter [EC2 Amazon-Instance-Typen](#).

Für jede Flotte geben Sie bis zu fünf EC2 Amazon-Instance-Typen an. Wenn Sie einen verwenden [Zuweisungsstrategie für Flotten](#) und mithilfe der AWS CLI oder der Amazon EMR-API einen Cluster erstellen, können Sie bis zu 30 EC2 Instance-Typen pro Instance-Flotte angeben. Amazon EMR wählt eine beliebige Kombination dieser EC2 Instance-Typen, um Ihre Zielkapazitäten zu erfüllen. Da Amazon EMR die Zielkapazität vollständig ausfüllen möchte, könnte es zu einer Übermenge kommen. Beispiel: Wenn zwei unerfüllte Einheiten vorhanden sind und Amazon EMR nur eine Instance mit einer Anzahl von fünf Einheiten bereitstellen kann, wird die Instance dennoch bereitgestellt. Dies bedeutet, dass die Zielkapazität im drei Einheiten überschritten wird.

Wenn Sie die Zielkapazität verringern, um die Größe eines laufenden Clusters zu ändern, versucht Amazon EMR, Anwendungsaufgaben abzuschließen und beendet Instances, um dem neuen Ziel zu entsprechen. Weitere Informationen finden Sie unter [Beendigung bei Aufgaben-Abschluss](#).

Start-Optionen

Für Spot Instances können Sie einen maximalen Spot-Preis für jeden Instance-Typ in einer Flotte angeben. Sie können den Preis entweder als Prozentsatz des On-Demand-Preises oder als einen bestimmten Betrag in US-Dollar festlegen. Amazon EMR stellt Spot Instances bereit, wenn der aktuelle Spot-Preis in einer Availability Zone unter Ihrem maximalen Spot-Preis liegt. Sie zahlen den Spot-Preis, nicht unbedingt den maximalen Spot-Preis.

 Note

Spot-Instances mit definierter Laufzeit (auch Spot-Blöcke genannt) stehen Neukunden ab dem 1. Juli 2021 nicht mehr zur Verfügung. Für Kunden, die diese Funktion bereits genutzt haben, werden wir Spot-Instances mit einer definierten Laufzeit bis zum 31. Dezember 2022 weiterhin unterstützen.

In Amazon EMR 5.12.1 und höher verfügbar, haben Sie die Möglichkeit, Spot- und On-Demand-Instance-Flotten mit optimierter Kapazitätszuweisung zu starten. Diese Option für die Zuweisungsstrategie kann in der alten Version AWS Management Console oder mithilfe der API `RunJobFlow` festgelegt werden. Beachten Sie, dass Sie die Zuweisungsstrategie in der neuen-Konsole nicht anpassen können. Für die Verwendung der Option „Zuweisungsstrategie“ sind zusätzliche Berechtigungen für Servicerollen erforderlich. Wenn Sie die standardmäßige Amazon-EMR-Servicerolle und die verwaltete Richtlinie ([EMR_DefaultRole](#) und `AmazonEMRServicePolicy_v2`) für den Cluster verwenden, sind die Berechtigungen für die Option Zuweisungsstrategie bereits enthalten. Wenn Sie nicht die standardmäßige Amazon-EMR-Servicerolle und die verwaltete Richtlinie verwenden, müssen Sie sie hinzufügen, um diese Option verwenden zu können. Siehe [Servicerolle für Amazon EMR \(EMR-Rolle\)](#).

Weitere Informationen zu Spot-Instances finden Sie unter [Spot-Instances](#) im EC2 Amazon-Benutzerhandbuch. Weitere Informationen zu On-Demand-Instances finden Sie unter [On-Demand-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Wenn Sie On-Demand-Instance-Flotten mit der Zuweisungsstrategie zum niedrigsten Preis starten möchten, haben Sie die Möglichkeit, Kapazitätsreservierungen zu verwenden. Optionen zur Kapazitätsreservierung können mithilfe der Amazon EMR API `RunJobFlow` festgelegt werden. Für Kapazitätsreservierungen sind zusätzliche Berechtigungen für Servicerollen erforderlich, die Sie hinzufügen müssen, um diese Optionen nutzen zu können. Siehe [Zuweisungsstrategie-Berechtigungen](#). Beachten Sie, dass Sie Kapazitätsreservierungen in der neuen Konsole nicht anpassen können.

Optionen für mehrere Subnetze (Availability Zones)

Wenn Sie Instance-Flotten verwenden, können Sie mehrere EC2 Amazon-Subnetze innerhalb einer VPC angeben, die jeweils einer anderen Availability Zone entsprechen. Wenn Sie EC2 -Classic verwenden, geben Sie Availability Zones explizit an. Amazon EMR identifiziert die beste Availability Zone zum Starten von Instances entsprechend den Spezifikationen Ihrer Flotte. Instances werden immer nur in einer Availability Zone bereitgestellt. Sie können private Subnetze oder öffentliche Subnetze auswählen, aber nicht beides zusammen. Die Subnetze, die Sie angeben, müssen sich in derselben VPC befinden.

Hauptknoten-Konfiguration

Da die Primär-Instance-Flotte nur eine einzelne Instance ist, unterscheidet sich ihre Konfiguration etwas von Core- und Task-Instance-Flotten. Wählen Sie entweder On-Demand oder Spot für die Primär-Instance-Flotte aus, da sie nur aus einer Instance besteht. Wenn Sie die Konsole verwenden,

um die Instance-Flotte zu erstellen, wird die Zielkapazität für die Kaufoption, die Sie auswählen, auf "1" festgelegt. Wenn Sie die verwenden AWS CLI, legen Sie je nach Bedarf immer entweder `TargetSpotCapacity` oder `TargetOnDemandCapacity` auf 1 fest. Sie können weiterhin bis zu fünf Instance-Typen für die primäre Instance-Flotte wählen (oder maximal 30, wenn Sie die Zuweisungsstrategie-Option für On-Demand- oder Spot Instances verwenden). Im Gegensatz zu Core- und Aufgaben-Instance-Flotten, für die Amazon EMR mehrere Instances mit verschiedenen Typen bereitstellen kann, wählt Amazon EMR einen einzelnen Instance-Typ zur Bereitstellung für die Primär-Instance-Flotte aus.

Zuweisungsstrategie für Flotten

Mit Amazon-EMR-Versionen 5.12.1 und höher können Sie die Option für die Zuweisungsstrategie mit On-Demand-Instances und Spot Instances für jeden Clusterknoten verwenden. Wenn Sie einen Cluster mithilfe der AWS CLI Amazon EMR-API oder der Amazon EMR-Konsole mit einer Zuweisungsstrategie erstellen, können Sie bis zu 30 EC2 Amazon-Instance-Typen pro Flotte angeben. Mit der standardmäßigen Amazon-EMR-Cluster-Instance-Flottenkonfiguration können Sie bis zu 5 Instance-Typen pro Flotte verwenden. Wir empfehlen Ihnen, die Option für die Zuweisungsstrategie zu verwenden, um eine schnellere Cluster-Bereitstellung, eine genauere Spot-Instance-Zuweisung und weniger Spot Instance-Unterbrechungen zu erzielen.

Themen

- [Zuweisungsstrategie mit On-Demand-Instances](#)
- [Zuweisungsstrategie mit Spot Instances](#)
- [Zuweisungsstrategie-Berechtigungen](#)
- [Erforderliche IAM-Berechtigungen für eine Zuweisungsstrategie](#)

Zuweisungsstrategie mit On-Demand-Instances

Die folgenden Zuweisungsstrategien sind für Ihre On-Demand-Instances verfügbar:

`lowest-price(Standard)`

Bei der Zuweisungsstrategie mit dem niedrigsten Preis werden On-Demand-Instances aus dem Pool mit dem niedrigsten Preis gestartet, der über verfügbare Kapazität verfügt. Wenn der Pool mit dem niedrigsten Preis keine verfügbare Kapazität hat, stammen die On-Demand-Instances aus dem Pool mit dem nächstniedrigsten Preis und verfügbarer Kapazität.

prioritized

Mit der priorisierten Zuweisungsstrategie können Sie für jeden Instance-Typ Ihrer Instance-Flotte einen Prioritätswert angeben. Amazon EMR startet Ihre On-Demand-Instances mit der höchsten Priorität. Wenn Sie diese Strategie verwenden, müssen Sie die Priorität für mindestens einen Instance-Typ konfigurieren. Wenn Sie den Prioritätswert für einen Instance-Typ nicht konfigurieren, weist Amazon EMR diesem Instance-Typ die niedrigste Priorität zu. Jede Instance-Flotte (primär, Core oder Task) in einem Cluster kann für einen bestimmten Instance-Typ einen anderen Prioritätswert haben.

Note

Wenn Sie die `capacity-optimized-prioritizedSpot`-Zuweisungsstrategie verwenden, wendet Amazon EMR bei der Festlegung von Prioritäten dieselben Prioritäten sowohl auf Ihre On-Demand-Instances als auch auf Spot-Instances an.

Zuweisungsstrategie mit Spot Instances

Für Spot Instances können Sie aus einer der folgenden Zuweisungsstrategien wählen:

price-capacity-optimized (empfohlen)

Bei der preis-kapazitätsoptimierten Zuweisungsstrategie werden Spot Instances aus den Spot Instance-Pools gestartet, die über die höchste verfügbare Kapazität und den niedrigsten Preis für die Anzahl der zu startenden Instances verfügen. Aus diesem Grund bietet die Strategie mit optimierter Preis- und Kapazitätsoptimierung in der Regel eine höhere Wahrscheinlichkeit, Spot-Kapazität zu erhalten, und führt zu niedrigeren Unterbrechungsraten. Dies ist die Standardstrategie für Amazon EMR-Versionen 6.10.0 und höher.

capacity-optimized

Die kapazitätsoptimierte Zuweisungsstrategie startet Spot Instances in den am meisten verfügbaren Pools mit der geringsten Wahrscheinlichkeit einer kurzfristigen Unterbrechung. Dies ist eine gute Option für Workloads, bei denen Unterbrechungen aufgrund von Neustarts von Aufgaben höhere Kosten verursachen. Dies ist die Standardstrategie für Amazon-EMR-Versionen 6.9.0 und niedriger.

capacity-optimized-prioritized

Mit der capacity-optimized-prioritized Zuweisungsstrategie können Sie für jeden Instance-Typ in Ihrer Instance-Flotte einen Prioritätswert angeben. Amazon EMR optimiert zunächst die Kapazität, berücksichtigt jedoch die Prioritäten des Instance-Typs nach bestem Wissen und Gewissen, z. B. wenn die Priorität die Fähigkeit der Flotte, optimale Kapazität bereitzustellen, nicht wesentlich beeinträchtigt. Wir empfehlen diese Option, wenn Sie Workloads haben, die möglichst wenig unterbrochen werden müssen und dennoch bestimmte Instance-Typen benötigt werden. Wenn Sie diese Strategie verwenden, müssen Sie die Priorität für mindestens einen Instance-Typ konfigurieren. Wenn Sie für keinen Instance-Typ eine Priorität konfigurieren, weist Amazon EMR diesem Instance-Typ den niedrigsten Prioritätswert zu. Jede Instance-Flotte (primär, Core oder Task) in einem Cluster kann für einen bestimmten Instance-Typ einen anderen Prioritätswert haben.

Note

Wenn Sie die priorisierte On-Demand-Zuweisungsstrategie verwenden, wendet Amazon EMR bei der Festlegung von Prioritäten den gleichen Prioritätswert sowohl auf Ihre On-Demand-Instances als auch auf Ihre Spot-Instances an.

diversified

Mit der diversifizierten Zuweisungsstrategie EC2 verteilt Amazon Spot-Instances auf alle Spot-Kapazitätspools.

lowest-price

Bei der preisgünstigsten Zuweisungsstrategie werden Spot Instances aus dem preisgünstigsten Pool mit verfügbarer Kapazität gestartet. Wenn der günstigste Pool keine verfügbare Kapazität aufweist, kommen die Spot Instances aus dem nächstgünstigsten Pool mit verfügbarer Kapazität. Wenn die Kapazität eines Pools knapp wird, bevor er die von Ihnen angeforderte Kapazität erfüllt, greift die EC2 Amazon-Flotte auf den Pool mit dem nächstniedrigsten Preis zurück, um Ihre Anfrage weiter zu bearbeiten. Damit die gewünschte Kapazität auf jeden Fall erreicht wird, erhalten Sie möglicherweise Spot-Instances aus mehreren Pools. Da diese Strategie nur den Instance-Preis und nicht die Kapazitätsverfügbarkeit berücksichtigt, kann es zu hohen Unterbrechungsraten kommen.

Zuweisungsstrategie-Berechtigungen

Die Option für die Zuweisungsstrategie erfordert mehrere IAM-Berechtigungen, die automatisch in der standardmäßigen Amazon-EMR-Servicerolle und der von Amazon EMR verwalteten Richtlinie (EMR_DefaultRole und AmazonEMRServicePolicy_v2) enthalten sind. Wenn Sie eine benutzerdefinierte Servicerolle oder eine verwaltete Richtlinie für Ihren Cluster verwenden, müssen Sie diese Berechtigungen hinzufügen, bevor Sie den Cluster erstellen. Weitere Informationen finden Sie unter [Zuweisungsstrategie-Berechtigungen](#).

Optionale Kapazitätsreservierungen auf Abruf (ODCRs) sind verfügbar, wenn Sie die Option für die On-Demand-Zuweisungsstrategie verwenden. Mit den Optionen zur Kapazitätsreservierung können Sie angeben, ob reservierte Kapazität zuerst für Amazon-EMR-Cluster verwendet werden soll. Sie können damit sicherstellen, dass Ihre kritischen Workloads die Kapazität nutzen, die Sie bereits mit Open oder Targeted ODCRs reserviert haben. Bei unkritischen Workloads können Sie in den Einstellungen für die Kapazitätsreservierung angeben, ob reservierte Kapazität verbraucht werden soll.

Kapazitätsreservierungen können nur von Instances verwendet werden, die ihren Attributen (Instance-Typ, Plattform und Availability Zone) entsprechen. Standardmäßig werden offene Kapazitätsreservierungen automatisch von Amazon EMR verwendet, wenn On-Demand-Instances bereitgestellt werden, die den Instance-Attributen entsprechen. Wenn Sie keine laufenden Instances haben, die den Attributen der Kapazitätsreservierungen entsprechen, bleiben diese ungenutzt, bis Sie eine Instance starten, die ihren Attributen entspricht. Wenn Sie beim Starten Ihres Clusters keine Kapazitätsreservierungen verwenden möchten, müssen Sie in den Startoptionen die Einstellung „Kapazitätsreservierung“ auf Keine setzen.

Sie können jedoch auch eine Kapazitätsreservierung für bestimmte Workloads festlegen. Auf diese Weise können Sie explizit steuern, welche Instances in der reservierten Kapazität ausgeführt werden dürfen. Weitere Informationen über On-Demand-Kapazitätsreservierungen finden Sie unter [Verwenden Sie Kapazitätsreservierungen mit Instance-Flotten in Amazon EMR](#).

Erforderliche IAM-Berechtigungen für eine Zuweisungsstrategie

Ihre [Servicerolle für Amazon EMR \(EMR-Rolle\)](#) benötigen zusätzliche Berechtigungen, um einen Cluster zu erstellen, der die Zuweisungsstrategieoption für On-Demand-Instance-Flotten oder Spot-Instance-Flotten verwendet.

Wir nehmen diese Berechtigungen automatisch in die standardmäßige Amazon-EMR-Servicerolle [EMR_DefaultRole](#) und die von Amazon EMR verwaltete Richtlinie [AmazonEMRServicePolicy_v2](#) auf.

Wenn Sie eine benutzerdefinierte Servicerolle oder eine verwaltete Richtlinie für Ihren Cluster verwenden, müssen Sie die folgenden Berechtigungen hinzufügen:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteLaunchTemplate",
        "ec2:CreateLaunchTemplate",
        "ec2:DescribeLaunchTemplates",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateFleet"
      ],
      "Resource": "*"
    }
  ]
}
```

Die folgenden Berechtigungen für Servicerollen sind erforderlich, um einen Cluster zu erstellen, der offene oder gezielte Kapazitätsreservierungen verwendet. Sie müssen diese Berechtigungen zusätzlich zu den Berechtigungen angeben, die für die Verwendung der Zuweisungsstrategie-Option erforderlich sind.

Example Richtliniendokument für Kapazitätsreservierungen für Servicerollen

Um offene Kapazitätsreservierungen verwenden zu können, müssen Sie die folgenden zusätzlichen Berechtigungen angeben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeCapacityReservations",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DeleteLaunchTemplateVersions"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

Example

Um gezielte Kapazitätsreservierungen verwenden zu können, müssen Sie die folgenden zusätzlichen Berechtigungen angeben.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeCapacityReservations",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DeleteLaunchTemplateVersions",
        "resource-groups:ListGroupResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Instance-Flotten für Ihren Cluster konfigurieren

Console

Um mit der Konsole einen Cluster mit Instanzflotten zu erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Clusterkonfiguration die Option Instance-Flotten aus.
4. Wählen Sie für jede Knotengruppe die Option Instance-Typ hinzufügen und wählen Sie bis zu 5 Instance-Typen für Primär- und Core-Instance-Flotten und bis zu fünfzehn Instance-Typen für Aufgaben-Instance-Flotten aus. Amazon EMR kann jede beliebige Kombination dieser Instance-Typen beim Starten des Clusters bereitstellen.
5. Wählen Sie unter jedem Knotengruppentyp das Drop-Down-Menü Aktionen neben jeder Instance aus, um diese Einstellungen zu ändern:

EBS-Volumes hinzufügen

Geben Sie EBS-Volumes an, die an den Instance-Typ angefügt werden sollen, nachdem Amazon EMR ihn bereitgestellt hat.

Gewichtete Kapazität bearbeiten

Ändern Sie diesen Wert für die Core-Knotengruppe auf eine beliebige Anzahl von Einheiten, die Ihren Anwendungen entspricht. Die Anzahl der virtuellen YARN-Kerne für jeden Flotten-Instance-Typ wird als standardmäßige gewichtete Kapazitätseinheiten verwendet. Sie können die gewichtete Kapazität für den Primärknoten nicht bearbeiten.

Den maximalen Spot-Preis bearbeiten

Geben Sie für jeden Instance-Typ in einer Flotte einen maximalen Spot-Preis an. Sie können den Preis entweder als Prozentsatz des On-Demand-Preises oder als einen bestimmten Betrag in US-Dollar festlegen. Amazon EMR stellt Spot Instances bereit, wenn der aktuelle Spot-Preis in einer Availability Zone unter Ihrem maximalen Spot-Preis liegt. Sie zahlen den Spot-Preis, nicht unbedingt den maximalen Spot-Preis.

6. Um optional Sicherheitsgruppen für Ihre Knoten hinzuzufügen, erweitern Sie EC2 Sicherheitsgruppen (Firewall) im Bereich Netzwerk und wählen Sie Ihre Sicherheitsgruppe für jeden Knotentyp aus.
7. Aktivieren Sie optional das Kontrollkästchen neben Zuweisungsstrategie anwenden, wenn Sie die Option Zuweisungsstrategie verwenden möchten, und wählen Sie die Zuweisungsstrategie aus, die Sie für die Spot Instances angeben möchten. Sie sollten diese Option nicht auswählen, wenn Ihre Amazon-EMR-Servicerolle nicht über die erforderlichen Berechtigungen verfügt. Weitere Informationen finden Sie unter [Zuweisungsstrategie für Flotten](#).
8. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
9. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um einen Cluster mit Instanzflotten mit den zu erstellen und zu starten AWS CLI, folgen Sie diesen Richtlinien:

- Zum Erstellen und Starten eines Clusters mit Instance-Flotten verwenden Sie den Befehl `create-cluster` zusammen mit `--instance-fleet`-Parametern.

- Um mehr Konfigurationsdetails der Instance-Flotten in einem Cluster zu erhalten, verwenden Sie den Befehl `list-instance-fleets`.
- AMLs Um einem Cluster, den Sie erstellen, mehrere benutzerdefinierte Amazon Linux-Benutzer hinzuzufügen, verwenden Sie die `CustomAmiId` Option für jede `InstanceType` Spezifikation. Sie können Instance-Flottenknoten mit mehreren Instance-Typen und mehreren benutzerdefinierten Instance-Typen konfigurieren AMLs , um Ihren Anforderungen zu entsprechen. Siehe [Beispiele: Erstellen eines Clusters mit der Instance-Flotten-Konfiguration](#).
- Wenn Sie die Zielkapazität für eine Instance-Flotte ändern möchten, verwenden Sie den Befehl `modify-instance-fleet`.
- Zum Hinzufügen einer Aufgaben-Instance-Flotte zu einem Cluster, dem noch keine Flotte zugewiesen wurde, verwenden Sie den Befehl `add-instance-fleet`.
- Mithilfe des `CustomAmiId` Arguments mit dem `add-instance-fleet` Befehl AMLs können der Task-Instance-Flotte mehrere benutzerdefinierte hinzugefügt werden. Siehe [Beispiele: Erstellen eines Clusters mit der Instance-Flotten-Konfiguration](#).
- Um die Option für die Zuweisungsstrategie bei der Erstellung einer Instance-Flotte zu verwenden, aktualisieren Sie die Servicerolle so, dass sie das Beispielrichtliniendokument im folgenden Abschnitt enthält.
- Um die Optionen für Kapazitätsreservierungen bei der Erstellung einer Instance-Flotte mit On-Demand-Zuweisungsstrategie zu verwenden, aktualisieren Sie die Servicerolle so, dass sie das Beispielrichtliniendokument im folgenden Abschnitt enthält.
- Die Instance-Flotten sind automatisch in der standardmäßigen EMR-Servicerolle und der von Amazon EMR verwalteten Richtlinie (`EMR_DefaultRole` und `AmazonEMRServicePolicy_v2`) enthalten. Wenn Sie eine benutzerdefinierte Servicerolle oder eine vom Kunden verwaltete Richtlinie für Ihren Cluster verwenden, müssen Sie die neuen Berechtigungen für die Zuweisungsstrategie im folgenden Abschnitt hinzufügen.

Beispiele: Erstellen eines Clusters mit der Instance-Flotten-Konfiguration

Die folgenden Beispiele zeigen `create-cluster`-Befehle mit einer Vielzahl von Optionen, die Sie kombinieren können.

Note

Wenn Sie noch nicht die standardmäßige Amazon EMR-Servicerolle und das EC2 Instanzprofil erstellt haben, erstellen Sie `aws emr create-default-roles` diese zunächst, bevor Sie den `create-cluster` Befehl verwenden.

Example Beispiel: On-Demand-Primär, On-Demand-Core mit individuellem Instance-Typ, Standard-VPC

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}']
```

Example Beispiel: Spot-Primär, Spot-Core mit individuellem Instance-Typ, Standard-VPC

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetSpotCapacity=1,\
    InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=1,\
    InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

Example Beispiel: Primärer On-Demand-Prozessor, gemischter Kern mit Einzelinstanztyp, einzelnes Subnetz EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c'] \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
    InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=2,TargetSpotCapacity=6,\
    InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=2}']
```

Example Beispiel: Primär auf Abruf, Spot-Core mit mehreren gewichteten Instance-Typen, Timeout für Spot, Subnetzbereich EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
  InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=11,\
  InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
  LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
```

Example Beispiel: Primäre On-Demand-Instanz, gemischter Kern- und Task-Instance-Typ mit mehreren gewichteten Instance-Typen, Timeout für Core-Spot-Instances, Subnetzbereich EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
  --instance-fleets \

  InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=8,TargetSpotCapacity=6,\
  InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
  LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON}'] \
    InstanceFleetType=TASK,TargetOnDemandCapacity=3,TargetSpotCapacity=3,\
  InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}']
```

Example Beispiel: Spot-Primär, kein Core oder Aufgabe, Amazon-EBS-Konfiguration, Standard-VPC

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetSpotCapacity=1,\
  LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=60,TimeoutAction=TERMINATE_CLUSTER}'] \
  InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,\
```

```
EbsConfiguration={EbsOptimized=true,EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=gp2,
\
SizeInGB=100}}, {VolumeSpecification={VolumeType=io1,SizeInGB=100,Iops=100},VolumesPerInstance=4}}]}
```

Example Beispiel: Mehrere benutzerdefinierte AMIs, mehrere Instance-Typen, primärer On-Demand-Instance-Typ, On-Demand-Core

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},\
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},\
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}']
```

Example Beispiel: Fügen Sie einem laufenden Cluster mit mehreren Instanztypen und mehreren benutzerdefinierten Instanztypen einen Task-Knoten hinzu AMIs

```
aws emr add-instance-fleet --cluster-id j-123456 --release-label Amazon EMR 5.3.1 \
--service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleet \
    InstanceFleetType=Task,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}',\
'{InstanceType=m6g.xlarge,CustomAmiId=ami-234567}']
```

Example Beispiel: Eine JSON-Konfigurationsdatei verwenden

Sie können Instance-Flotten-Parameter in einer JSON-Datei konfigurieren und dann auf die JSON-Datei als einzigen Parameter für Instance-Flotten verweisen. Mit dem folgenden Befehl wird z. B. auf die JSON-Konfigurationsdatei *my-fleet-config.json* verwiesen:

```
aws emr create-cluster --release-label emr-5.30.0 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
--instance-fleets file://my-fleet-config.json
```

In der *my-fleet-config.json* Datei werden Primär-, Kern- und Taskinstanzflotten angegeben, wie im folgenden Beispiel gezeigt. Die Core-Instance-Flotte verwendet einen maximalen Spot-Preis (BidPrice) als Prozentsatz von On-Demand, während die Task- und Primärinstance-Flotten einen maximalen Spot-Preis (BidPriceAsPercentageofOnDemandPrice) als Zeichenfolge in USD verwenden.

```
[
  {
    "Name": "Masterfleet",
    "InstanceFleetType": "MASTER",
    "TargetSpotCapacity": 1,
    "LaunchSpecifications": {
      "SpotSpecification": {
        "TimeoutDurationMinutes": 120,
        "TimeoutAction": "SWITCH_TO_ON_DEMAND"
      }
    },
    "InstanceTypeConfigs": [
      {
        "InstanceType": "m5.xlarge",
        "BidPrice": "0.89"
      }
    ]
  },
  {
    "Name": "Corefleet",
    "InstanceFleetType": "CORE",
    "TargetSpotCapacity": 1,
    "TargetOnDemandCapacity": 1,
    "LaunchSpecifications": {
      "OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions": {
          "UsageStrategy": "use-capacity-reservations-first",
          "CapacityReservationResourceGroupArn": "String"
        }
      },
      "SpotSpecification": {
        "AllocationStrategy": "capacity-optimized",
        "TimeoutDurationMinutes": 120,
        "TimeoutAction": "TERMINATE_CLUSTER"
      }
    }
  }
]
```



```

    },
    "InstanceTypeConfigs": [
        {
            "InstanceType": "m5.xlarge",
            "BidPriceAsPercentageOfOnDemandPrice": 100
        }
    ]
},
{
    "Name": "Taskfleet",
    "InstanceFleetType": "TASK",
    "TargetSpotCapacity": 1,
    "LaunchSpecifications": {
        "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price",
            "CapacityReservationOptions": {
                "CapacityReservationPreference": "none"
            }
        },
        "SpotSpecification": {
            "TimeoutDurationMinutes": 120,
            "TimeoutAction": "TERMINATE_CLUSTER"
        }
    },
    "InstanceTypeConfigs": [
        {
            "InstanceType": "m5.xlarge",
            "BidPrice": "0.89"
        }
    ]
}
]

```

Zielkapazitäten für eine Instance-Flotte ändern

Verwenden Sie den Befehl `modify-instance-fleet`, um neue Zielkapazitäten für eine Instance-Flotte anzugeben. Sie müssen die Cluster-ID und die Instance-Flotten-ID angeben. Verwenden Sie den `list-instance-fleets` Befehl, um die Instance-Flotte IDs abzurufen.

```
aws emr modify-instance-fleet --cluster-id <cluster-id> \
    --instance-fleet \
```

```
InstanceFleetId='<instance-fleet-id>',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

Eine Aufgaben-Instance-Flotte zu einem Cluster hinzufügen

Wenn ein Cluster nur über Primär- und Core-Instance-Flotten verfügt, können Sie den Befehl `add-instance-fleet` verwenden, um eine Aufgaben-Instance-Flotte hinzuzufügen. Sie können nur diesen Befehl verwenden, um Aufgaben-Instance-Flotten hinzuzufügen.

```
aws emr add-instance-fleet --cluster-id <cluster-id>
  --instance-fleet \
    InstanceFleetType=TASK,TargetSpotCapacity=1,\
  LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=20,TimeoutAction=TERMINATE_CLUSTER,\
  \
  InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']}
```

Konfigurationsdetails der Instance-Flotten in einem Cluster abrufen

Verwenden Sie den Befehl `list-instance-fleets`, um Konfigurationsdetails der Instance-Flotten in einem Cluster abzurufen. Der Befehl erfordert die Eingabe einer Cluster-ID. Das folgende Beispiel zeigt den Befehl und die Ausgabe für einen Cluster mit einer Primär-Aufgaben-Instance-Gruppe und einer Core-Aufgaben-Instance-Gruppe. Die vollständige Antwortsyntax finden Sie [ListInstanceFleets](#) in der Amazon EMR API-Referenz.

```
list-instance-fleets --cluster-id <cluster-id>
```

```
{
  "InstanceFleets": [
    {
      "Status": {
        "Timeline": {
          "ReadyDateTime": 1488759094.637,
          "CreationDateTime": 1488758719.817
        },
        "State": "RUNNING",
        "StateChangeReason": {
          "Message": ""
        }
      },
      "ProvisionedSpotCapacity": 6,
      "Name": "CORE",
```

```

    "InstanceFleetType": "CORE",
    "LaunchSpecifications": {
        "SpotSpecification": {
            "TimeoutDurationMinutes": 60,
            "TimeoutAction": "TERMINATE_CLUSTER"
        }
    },
    "ProvisionedOnDemandCapacity": 2,
    "InstanceTypeSpecifications": [
        {
            "BidPrice": "0.5",
            "InstanceType": "m5.xlarge",
            "WeightedCapacity": 2
        }
    ],
    "Id": "if-1ABC2DEFGHIJ3"
},
{
    "Status": {
        "Timeline": {
            "ReadyDateTime": 1488759058.598,
            "CreationDateTime": 1488758719.811
        },
        "State": "RUNNING",
        "StateChangeReason": {
            "Message": ""
        }
    },
    "ProvisionedSpotCapacity": 0,
    "Name": "MASTER",
    "InstanceFleetType": "MASTER",
    "ProvisionedOnDemandCapacity": 1,
    "InstanceTypeSpecifications": [
        {
            "BidPriceAsPercentageOfOnDemandPrice": 100.0,
            "InstanceType": "m5.xlarge",
            "WeightedCapacity": 1
        }
    ],
    "Id": "if-2ABC4DEFGHIJ4"
}
]
}

```

Neukonfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster

Mit Amazon EMR Version 5.21.0 und höher können Sie Cluster-Anwendungen neu konfigurieren und zusätzliche Konfigurationsklassifizierungen für jede Instance-Flotte in einem laufenden Cluster angeben. Dazu können Sie die AWS Befehlszeilenschnittstelle (AWS CLI) oder das SDK verwenden. AWS

Sie können den Status einer Instance-Flotte verfolgen, indem Sie sich die CloudWatch Ereignisse ansehen. Weitere Informationen finden Sie unter [Ereignisse zur Neukonfiguration der Instance-Flotte](#).

Note

Sie können nur das Cluster-Konfigurationsobjekt überschreiben, das bei der Clustererstellung angegeben wurde. Weitere Informationen zu Konfigurationsobjekten finden Sie unter [RunJobFlow Anforderungssyntax](#). Wenn es Unterschiede zwischen der vorhandenen Konfiguration und der von Ihnen bereitgestellten Datei gibt, setzt Amazon EMR manuell geänderte Konfigurationen, z. B. Konfigurationen, die Sie geändert haben, während Sie über SSH mit Ihrem Cluster verbunden waren, auf die Cluster-Standard Einstellungen für die angegebene Instance-Flotte zurück.

Wenn Sie eine Rekonfigurationsanfrage über die Amazon EMR-Konsole, die AWS Befehlszeilenschnittstelle (AWS CLI) oder das AWS SDK einreichen, überprüft Amazon EMR die vorhandene Cluster-Konfigurationsdatei. Wenn es Unterschiede zwischen der vorhandenen Konfiguration und der von Ihnen bereitgestellten Datei gibt, initiiert Amazon EMR Neukonfigurationsaktionen, startet einige Anwendungen neu und setzt alle manuell geänderten Konfigurationen, wie Konfigurationen, die Sie geändert haben, während Sie über SSH mit Ihrem Cluster verbunden waren, auf die Cluster-Standard Einstellungen für die angegebene Instance-Flotte zurück.

Verhalten bei der Neukonfiguration

Die Neukonfiguration überschreibt die Cluster-Konfiguration mit dem neu übermittelten Konfigurationssatz und kann Konfigurationsänderungen überschreiben, die außerhalb der Rekonfigurations-API vorgenommen wurden.

Amazon EMR folgt einem fortlaufenden Prozess zur Neukonfiguration von Instances in der Task- und Core-Instance-Flotte. Nur ein Prozentsatz der Instances für einen einzelnen Instance-Typ wird

gleichzeitig geändert und neu gestartet. Wenn Ihre Instance-Flotte über mehrere verschiedene Instance-Typkonfigurationen verfügt, werden diese parallel neu konfiguriert.

Rekonfigurationen werden auf der Ebene deklariert. [InstanceTypeConfig](#) Ein visuelles Beispiel finden Sie unter. [Konfigurieren Sie eine Instanzflotte neu](#) Sie können Rekonfigurationsanfragen, die aktualisierte Konfigurationseinstellungen für einen oder mehrere Instance-Typen enthalten, innerhalb einer einzigen Anfrage einreichen. Sie müssen alle Instance-Typen, die Teil Ihrer Instance-Flotte sind, in die Änderungsanforderung aufnehmen. Instance-Typen mit ausgefüllten Konfigurationsfeldern werden jedoch neu konfiguriert, während andere `InstanceTypeConfig` Instances in der Flotte unverändert bleiben. Eine Neukonfiguration gilt nur dann als erfolgreich, wenn alle Instances der angegebenen Instance-Typen die Neukonfiguration abgeschlossen haben. Wenn eine Instance nicht neu konfiguriert werden kann, kehrt die gesamte Instance-Flotte automatisch zur letzten bekannten stabilen Konfiguration zurück.

Einschränkungen

Beachten Sie bei der Neukonfiguration einer Instance-Flotte in einem laufenden Cluster die folgenden Einschränkungen:

- Anwendungen, die nicht von Yarn stammen, können beim Neustart fehlschlagen oder Clusterprobleme verursachen, insbesondere wenn die Anwendungen nicht richtig konfiguriert sind. Bei Clustern, die sich der maximalen Speicher- und CPU-Auslastung nähern, können nach dem Neustart Probleme auftreten. Dies gilt insbesondere für die primäre Instance-Flotte. Konsultieren Sie den [Beheben Sie Fehler bei der Neukonfiguration der Instance-Flotte](#) Abschnitt.
- Größenänderungen und Rekonfigurationsvorgänge erfolgen nicht parallel. Anfragen zur Neukonfiguration warten auf eine fortlaufende Größenänderung und umgekehrt.
- Größenänderungen und Rekonfigurationsvorgänge erfolgen nicht parallel. Anfragen zur Neukonfiguration warten auf eine fortlaufende Größenänderung und umgekehrt.
- Nach der Neukonfiguration einer Instance-Flotte startet Amazon EMR die Anwendungen neu, damit die neuen Konfigurationen wirksam werden können. Auftragsfehler und anderes unerwartetes Anwendungsverhalten kann auftreten, wenn die Anwendungen während der Rekonfiguration genutzt werden.
- Wenn eine Neukonfiguration für eine Instance-Typ-Konfiguration unter einer Instance-Flotte fehlschlägt, setzt Amazon EMR die Konfigurationsparameter für die gesamte Instance-Flotte auf die vorherige funktionierende Version zurück, sendet Ereignisse aus und aktualisiert Statusdetails. Wenn auch der Reversionsprozess fehlschlägt, müssen Sie eine neue `ModifyInstanceFleet` Anfrage einreichen, um die Instance-Flotte aus dem Status wiederherzustellen. `ARRESTED`

Fehler bei der Wiederherstellung führen zu [Neukonfigurationen der Instance-Flotte und zu Statusänderungen](#).

- Rekonfigurationsanforderungen für Phoenix-Konfigurationsklassifizierungen werden nur in Amazon EMR ab Version 5.23.0, nicht jedoch in den Amazon-EMR-Versionen 5.21.0 und 5.22.0 unterstützt.
- Rekonfigurationsanfragen für HBase Konfigurationsklassifizierungen werden nur in Amazon EMR Version 5.30.0 und höher unterstützt und werden in den Amazon EMR-Versionen 5.23.0 bis 5.29.0 nicht unterstützt.
- Die Neukonfiguration der hdfs-encryption-zones-Klassifizierung oder einer der Hadoop KMS-Konfigurationsklassifizierungen auf einem ausgeführten Cluster wird in Amazon EMR nicht unterstützt.
- Amazon EMR unterstützt derzeit bestimmte Rekonfigurationsanfragen für den YARN-Kapazitätsplaner nicht, die einen Neustart des YARN erfordern. ResourceManager Beispielsweise können Sie eine Warteschlange nicht vollständig entfernen.
- Wenn YARN neu gestartet werden muss, werden in der Regel alle laufenden YARN-Jobs beendet und gehen verloren. Dies kann zu Verzögerungen bei der Datenverarbeitung führen. Um YARN-Jobs während eines YARN-Neustarts auszuführen, können Sie entweder einen Amazon EMR-Cluster mit mehreren Primärknoten erstellen oder `yarn.resourcemanager.recovery.enabled` in Ihrer Yarn-Site-Konfigurationsklassifizierung auf `true` setzen. [Weitere Informationen zur Verwendung mehrerer Master-Knoten finden Sie unter YARN für hohe Verfügbarkeit. ResourceManager](#)

Konfigurieren Sie eine Instanzflotte neu

Using the AWS CLI

Verwenden Sie den `modify-instance-fleet` Befehl, um eine neue Konfiguration für eine Instance-Flotte in einem laufenden Cluster anzugeben.

 Note

Ersetzen Sie in den folgenden Beispielen `j-2 AL4 XXXXXX5 T9` durch Ihre Cluster-ID und `if-1xxxxxxx9` durch Ihre Instance-Flotte-ID.

Beispiel — Ersetzen Sie eine Konfiguration für eine Instance-Flotte

⚠ Warning

Geben Sie alle `InstanceTypeConfig` Felder an, die Sie beim Start verwendet haben. Wenn Sie Felder nicht angeben, kann dies dazu führen, dass die Spezifikationen, die Sie beim Start angegeben haben, überschrieben werden. Eine Liste finden Sie unter [InstanceTypeConfig](#)

Das folgende Beispiel verweist auf eine JSON-Konfigurationsdatei namens `instanceFleet.json`, um die Eigenschaft des NodeManager YARN-Festplattenintegritätsprüfers für eine Instance-Flotte zu bearbeiten.

Änderung der Instanzflotte (JSON)

1. Bereiten Sie Ihre Konfigurationsklassifizierung vor und speichern Sie sie als `InstanceFleet.json` in demselben Verzeichnis, in dem Sie den Befehl ausführen werden.

```
{
  "InstanceFleetId": "if-1xxxxxxx9",
  "InstanceTypeConfigs": [
    {
      "InstanceType": "m5.xlarge",
      other InstanceTypeConfig fields
      "Configurations": [
        {
          "Classification": "yarn-site",
          "Properties": {
            "yarn.nodemanager.disk-health-checker.enable": "true",
            "yarn.nodemanager.disk-health-checker.max-disk-
utilization-per-disk-percentage": "100.0"
          }
        }
      ]
    },
    {
      "InstanceType": "r5.xlarge",
      other InstanceTypeConfig fields
      "Configurations": [
        {
          "Classification": "yarn-site",
          "Properties": {
            "yarn.nodemanager.disk-health-checker.enable": "false",
```

```

        "yarn.nodemanager.disk-health-checker.max-disk-
utilization-per-disk-percentage":"70.0"
    }
}
]
}
]

```

2. Führen Sie den folgenden Befehl aus.

```

aws emr modify-instance-fleet \
--cluster-id j-2AL4XXXXXX5T9 \
--region us-west-2 \
--instance-fleet instanceFleet.json

```

Beispiel — Fügen Sie einer Instance-Flotte eine Konfiguration hinzu

Wenn Sie einem Instance-Typ eine Konfiguration hinzufügen möchten, müssen Sie alle zuvor angegebenen Konfigurationen für diesen Instance-Typ in Ihre neue ModifyInstanceFleet Anfrage aufnehmen. Andernfalls werden die zuvor für die angegebenen Konfigurationen entfernt.

Das folgende Beispiel fügt eine Eigenschaft für den NodeManager virtuellen YARN-Memory-Checker hinzu. Die Konfiguration enthält auch zuvor angegebene Werte für den NodeManager YARN-Festplattenintegritätsprüfer, sodass die Werte nicht überschrieben werden.

1. Bereiten Sie den folgenden Inhalt in instanceFleet.json vor und speichern Sie ihn in demselben Verzeichnis, in dem Sie den Befehl ausführen werden.

```

{
  "InstanceFleetId": "if-1xxxxxxxx9",
  "InstanceTypeConfigs": [
    {
      "InstanceType": "m5.xlarge",
      other InstanceTypeConfig fields
      "Configurations": [
        {
          "Classification": "yarn-site",
          "Properties": {
            "yarn.nodemanager.disk-health-checker.enable":"true",
            "yarn.nodemanager.disk-health-checker.max-disk-
utilization-per-disk-percentage":"100.0",

```



```

        "yarn.nodemanager.vmem-check-enabled": "true",
        "yarn.nodemanager.vmem-pmem-ratio": "3.0"
      }
    }
  ],
},
{
  "InstanceType": "r5.xlarge",
  other InstanceTypeConfig fields
  "Configurations": [
    {
      "Classification": "yarn-site",
      "Properties": {
        "yarn.nodemanager.disk-health-checker.enable": "false",
        "yarn.nodemanager.disk-health-checker.max-disk-
utilization-per-disk-percentage": "70.0"
      }
    }
  ]
}
]
}

```

2. Führen Sie den folgenden Befehl aus.

```

aws emr modify-instance-fleet \
--cluster-id j-2AL4XXXXXX5T9 \
--region us-west-2 \
--instance-fleet instanceFleet.json

```

using the Java SDK

Note

Ersetzen Sie in den folgenden Beispielen *j-2 AL4 XXXXXX5 T9* durch Ihre Cluster-ID und *if-1xxxxxxx9* durch Ihre Instance-Flotte-ID.

Der folgende Codeausschnitt bietet eine neue Konfiguration für eine Instance-Flotte, die das AWS SDK for Java verwendet.

```

AWSCredentials credentials = new BasicAWSCredentials("access-key", "secret-key");
AmazonElasticMapReduce emr = new AmazonElasticMapReduceClient(credentials);

Map<String,String> hiveProperties = new HashMap<String,String>();
hiveProperties.put("hive.join.emit.interval","1000");
hiveProperties.put("hive.merge.mapfiles","true");

Configuration newConfiguration = new Configuration()
    .withClassification("hive-site")
    .withProperties(hiveProperties);

List<InstanceTypeConfig> instanceTypeConfigList = new ArrayList<>();

for (InstanceTypeConfig instanceTypeConfig : currentInstanceTypeConfigList) {
    instanceTypeConfigList.add(new InstanceTypeConfig()
        .withInstanceType(instanceTypeConfig.getInstanceType())
        .withBidPrice(instanceTypeConfig.getBidPrice())
        .withWeightedCapacity(instanceTypeConfig.getWeightedCapacity())
        .withConfigurations(newConfiguration)
    );
}

InstanceFleetModifyConfig instanceFleetModifyConfig = new
    InstanceFleetModifyConfig()
        .withInstanceFleetId("if-1xxxxxxxx9")
        .withInstanceTypeConfigs(instanceTypeConfigList);

ModifyInstanceFleetRequest modifyInstanceFleetRequest = new
    ModifyInstanceFleetRequest()
        .withInstanceFleet(instanceFleetModifyConfig)
        .withClusterId("j-2AL4XXXXXX5T9");

emrClient.modifyInstanceFleet(modifyInstanceFleetRequest);

```

Beheben Sie Fehler bei der Neukonfiguration der Instance-Flotte

Wenn der Rekonfigurationsprozess für einen Instance-Typ innerhalb einer Instance-Flotte fehlschlägt, macht Amazon EMR die laufende Neukonfiguration rückgängig und protokolliert eine Fehlermeldung unter Verwendung von Ereignissen. Amazon CloudWatch Das Ereignis bietet eine kurze Zusammenfassung des Neukonfigurationsfehlers. Es listet die Instances auf, bei denen die

Rekonfiguration fehlgeschlagen ist, sowie die entsprechenden Fehlermeldungen. Das Folgende ist ein Beispiel für eine Fehlermeldung.

```
Amazon EMR couldn't revert the instance fleet if-1xxxxxxx9 in the Amazon EMR cluster j-2AL4XXXXXX5T9 (ExampleClusterName) to the previously successful configuration at 2021-01-01 00:00 UTC. The reconfiguration reversion failed because of Instance i-xxxxxxx1, i-xxxxxxx2, i-xxxxxxx3 failed with message "This is an example failure message"...
```

Um auf die Protokolle zur Knotenbereitstellung zuzugreifen

Verwenden Sie SSH, um eine Verbindung zu dem Knoten herzustellen, auf dem die Neukonfiguration fehlgeschlagen ist. Anweisungen finden Sie unter [Connect zu Ihrer Linux-Instance](#) in der Amazon Elastic Compute Cloud herstellen.

Accessing logs by connecting to a node

1. Navigieren Sie zum folgenden Verzeichnis, das die Protokolldateien für die Knotenbereitstellung enthält.

```
/mnt/var/log/provision-node/
```

2. Öffnen Sie das Unterverzeichnis reports und suchen Sie nach dem Node Provisioning Report für Ihre Rekonfiguration. Das Berichtsverzeichnis organisiert die Protokolle nach Versionsnummer der Rekonfiguration, UUID (Universally Unique Identifier), EC2 Amazon-Instance-IP-Adresse und Zeitstempel. Jeder Bericht ist eine komprimierte YAML-Datei, die detaillierte Informationen über den Rekonfigurationsprozess enthält. Im Folgenden finden Sie ein Beispiel für einen Berichts-Dateiname und einen Pfad.

```
/reports/2/ca598xxx-cxxx-4xxx-bxxx-6dbxxxxxxxxxx/ip-10-73-xxx-xxx.ec2.internal/202104061715.yaml.gz
```

3. Sie können einen Bericht mit einem Dateibetrachter wie zless untersuchen, wie im folgenden Beispiel gezeigt.

```
zless 202104061715.yaml.gz
```

Accessing logs using Amazon S3

Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>. Öffnen Sie den Amazon-S3-Bucket, den Sie angegeben haben, als Sie den Cluster für die Archivierung von Protokolldateien in konfiguriert haben.

1. Navigieren Sie zum folgenden Ordner, der die Protokolldateien für die Knotenbereitstellung enthält:

```
amzn-s3-demo-bucket/elasticmapreduce/cluster id/node/instance id/provision-node/
```

2. Öffnen Sie den Berichtsordner und suchen Sie nach dem Bericht zur Knotenbereitstellung für Ihre Neukonfiguration. Der Berichtsordner organisiert die Protokolle nach Versionsnummer der Rekonfiguration, UUID (Universally Unique Identifier), EC2 Amazon-Instance-IP-Adresse und Zeitstempel. Jeder Bericht ist eine komprimierte YAML-Datei, die detaillierte Informationen über den Rekonfigurationsprozess enthält. Im Folgenden finden Sie ein Beispiel für einen Berichts-Dateiname und einen Pfad.

```
/reports/2/ca598xxx-cxxx-4xxx-bxxx-6dbxxxxxxxxxx/ip-10-73-xxx-  
xxx.ec2.internal/202104061715.yaml.gz
```

Um eine Protokolldatei anzuzeigen, können Sie sie als Textdatei von Amazon S3 auf Ihren lokalen Computer herunterladen. Anweisungen finden Sie unter [Objekt herunterladen](#).

Jede Protokolldatei enthält einen detaillierten Bereitstellungsbericht für die zugehörige Rekonfiguration. Um Informationen zu Fehlermeldungen zu finden, können Sie nach der `err`-Protokollebene eines Berichts suchen. Das Berichtsformat hängt von der Amazon-EMR-Version auf Ihrem Cluster ab. Das folgende Beispiel zeigt Fehlerinformationen für die Amazon EMR-Release-Versionen 5.32.0 und 6.2.0 und höher, wobei das folgende Format verwendet wird:

```
- level: err  
  message: 'Example detailed error message.'  
  source: Puppet  
  tags:  
    - err  
  time: '2021-01-01 00:00:00.000000 +00:00'  
  file:
```

```
line:
```

Verwenden Sie Kapazitätsreservierungen mit Instance-Flotten in Amazon EMR

Um On-Demand-Instance-Flotten mit Optionen für Kapazitätsreservierungen zu starten, fügen Sie zusätzliche Servicerollenberechtigungen hinzu, die für die Nutzung von Kapazitätsreservierungsoptionen erforderlich sind. Da Optionen zur Kapazitätsreservierung zusammen mit der On-Demand-Zuweisungsstrategie verwendet werden müssen, müssen Sie auch die für die Zuweisungsstrategie erforderlichen Berechtigungen in Ihre Servicerolle und verwaltete Richtlinie aufnehmen. Weitere Informationen finden Sie unter [Zuweisungsstrategie-Berechtigungen](#).

Amazon EMR unterstützt sowohl offene als auch gezielte Kapazitätsreservierungen. Die folgenden Themen zeigen Konfigurationen von Instance-Flotten, die Sie zusammen mit der RunJobFlow-Aktion oder dem `create-cluster`-Befehl verwenden können, um Instance-Flotten mithilfe von On-Demand-Kapazitätsreservierungen zu starten.

Offene Kapazitätsreservierungen nach bestmöglichem Bemühen verwenden

Wenn die On-Demand-Instances des Clusters den in Ihrem Konto verfügbaren Attributen der offenen Kapazitätsreservierungen (Instance-Typ, Plattform, Tenancy und Availability Zone) entsprechen, werden die Kapazitätsreservierungen automatisch angewendet. Es kann jedoch nicht garantiert werden, dass Ihre Kapazitätsreservierungen genutzt werden. Für die Bereitstellung des Clusters bewertet Amazon EMR alle in der Startanforderung angegebenen Instance-Pools und verwendet den Pool mit dem niedrigsten Preis, der über ausreichend Kapazität verfügt, um alle angeforderten Core-Knoten zu starten. Verfügbare offene Kapazitätsreservierungen, die dem Instance-Pool entsprechen, werden automatisch angewendet. Wenn verfügbare offene Kapazitätsreservierungen nicht mit dem Instance-Pool übereinstimmen, bleiben sie ungenutzt.

Sobald die Core-Knoten bereitgestellt sind, wird die Availability Zone ausgewählt und repariert. Amazon EMR stellt Aufgabenknoten in Instance-Pools bereit, beginnend mit den günstigsten zuerst, in der ausgewählten Availability Zone, bis alle Aufgabenknoten bereitgestellt sind. Verfügbare offene Kapazitätsreservierungen, die den Instance-Pools entsprechen, werden automatisch angewendet.

Im Folgenden finden Sie Anwendungsfälle der Amazon-EMR-Kapazitätszuweisungslogik für die Nutzung offener Kapazitätsreservierungen nach bestem Wissen.

Beispiel 1: Der Instance-Pool mit dem niedrigsten Preis in der Startanfrage verfügt über verfügbare offene Kapazitätsreservierungen

In diesem Fall führt Amazon EMR Kapazität im Instance-Pool mit On-Demand-Instances mit dem niedrigsten Preis ein. Ihre verfügbaren offenen Kapazitätsreservierungen in diesem Instance-Pool werden automatisch verwendet.

On-Demand-Strategie	Niedrigster Preis		
Angeforderte Kapazität	100		
Instance-Typ	c5.xlarge	m5.xlarge	r5.xlarge
Verfügbare offene Kapazitätsreservierung	150	100	100
Preis auf Abruf	\$	\$\$	\$\$\$
Bereitgestellte Instances	100	-	-
Offene Kapazität sreservierung verwendet	100	-	-
Verfügbare offene Kapazitätsreservierung	50	100	100

Nachdem die Instance-Flotte gestartet wurde, können Sie [describe-capacity-reservations](#) ausführen, um zu sehen, wie viele ungenutzte Kapazitätsreservierungen verbleiben.

Beispiel 2: Für den Instance-Pool mit dem niedrigsten Preis in der Startanfrage sind keine offenen Kapazitätsreservierungen verfügbar

In diesem Fall führt Amazon EMR Kapazität im Instance-Pool mit On-Demand-Instances mit dem niedrigsten Preis ein. Ihre offenen Kapazitätsreservierungen bleiben jedoch ungenutzt.

On-Demand-Strategie	Niedrigster Preis

Angeforderte Kapazität	100		
Instance-Typ	c5.xlarge	m5.xlarge	r5.xlarge
Verfügbare offene Kapazitätsreservierung	-	-	100
Preis auf Abruf	\$	\$\$	\$\$\$
Bereitgestellte Instances	100	-	-
Offene Kapazitätsreservierung verwendet	-	-	-
Verfügbare offene Kapazitätsreservierung	-	-	100

Konfigurieren Sie Instance-Flotten so, dass offene Kapazitätsreservierungen nach bestem Wissen und Gewissen verwendet werden

Wenn Sie die `RunJobFlow`-Aktion verwenden, um einen auf Instance-Flotten basierenden Cluster zu erstellen, legen Sie für die On-Demand-Zuweisungsstrategie die Optionen `lowest-price` und `CapacityReservationPreference` für Kapazitätsreservierungen auf `open` fest. Wenn Sie dieses Feld leer lassen, setzt Amazon EMR alternativ die Kapazitätsreservierungspräferenz der On-Demand-Instance standardmäßig auf `open`.

```
"LaunchSpecifications":
  {"OnDemandSpecification": {
    "AllocationStrategy": "lowest-price",
    "CapacityReservationOptions":
      {
        "CapacityReservationPreference": "open"
      }
  }
}
```

}

Sie können auch die Amazon-EMR-CLI verwenden, um mithilfe von offenen Kapazitätsreservierungen einen auf Instance-Flotten basierenden Cluster zu erstellen.

```
aws emr create-cluster \
  --name 'open-ODCR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets
  InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarge}'] \
  \
  InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarge}',
  '{InstanceType=m5.xlarge}',{InstanceType=r5.xlarge}'],\
  LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-price,CapacityReservationOptions={CapacityReservationPreference=open}}'}
```

Wobei gilt,

- `open-ODCR-cluster` wird durch den Namen des Clusters ersetzt, der offenen Kapazitätsreservierungen verwendet.
- `subnet-22XXXX01` wird durch die Subnetz-ID ersetzt.

Zuerst offene Kapazitätsreservierungen verwenden

Sie können sich dafür entscheiden, bei der Bereitstellung eines Amazon-EMR-Clusters die Zuweisungsstrategie mit dem niedrigsten Preis außer Kraft zu setzen und zuerst verfügbare offene Kapazitätsreservierungen zu verwenden. In diesem Fall bewertet Amazon EMR alle Instance-Pools mit Kapazitätsreservierungen, die in der Startanfrage angegeben wurden, und verwendet den Pool mit dem niedrigsten Preis, der über ausreichend Kapazität verfügt, um alle angeforderten Core-Knoten zu starten. Wenn keiner der Instance-Pools mit Kapazitätsreservierungen über ausreichend Kapazität für die angeforderten Core-Knoten verfügt, greift Amazon EMR auf den im vorherigen Thema beschriebenen bestmöglichen Fall zurück. Das heißt, Amazon EMR bewertet alle in der Startanfrage angegebenen Instance-Pools neu und verwendet den Pool mit dem niedrigsten Preis, der über ausreichende Kapazität verfügt, um alle angeforderten Core-Knoten zu starten. Verfügbare offene Kapazitätsreservierungen, die dem Instance-Pool entsprechen, werden automatisch

angewendet. Wenn verfügbare offene Kapazitätsreservierungen nicht mit dem Instance-Pool übereinstimmen, bleiben sie ungenutzt.

Sobald die Core-Knoten bereitgestellt sind, wird die Availability Zone ausgewählt und repariert. Amazon EMR stellt Aufgabenknoten in der ausgewählten Availability Zone in Instance-Pools mit Kapazitätsreservierungen bereit, beginnend mit den günstigsten zuerst, bis alle Aufgabenknoten bereitgestellt sind. Amazon EMR verwendet zuerst die verfügbaren offenen Kapazitätsreservierungen, die für jeden Instance-Pool in der ausgewählten Availability Zone verfügbar sind, und verwendet nur bei Bedarf die niedrigste Preisstrategie, um alle verbleibenden Aufgabenknoten bereitzustellen.

Im Folgenden finden Sie Anwendungsfälle der Amazon-EMR-Kapazitätszuweisungslogik, bei der zuerst offene Kapazitätsreservierungen verwendet werden.

Beispiel 1: Der Instance-Pool mit verfügbaren offenen Kapazitätsreservierungen in der Startanfrage verfügt über ausreichend Kapazität für Core-Knoten

In diesem Fall startet Amazon EMR Kapazität im Instance-Pool mit verfügbaren offenen Kapazitätsreservierungen, unabhängig vom Preis des Instance-Pools. Daher werden Ihre offenen Kapazitätsreservierungen wann immer möglich genutzt, bis alle Core-Knoten bereitgestellt sind.

On-Demand-Strategie	Niedrigster Preis		
Angeforderte Kapazität	100		
Nutzungsstrategie	use-capacity-reservations-first		
Instance-Typ	c5.xlarge	m5.xlarge	r5.xlarge
Verfügbare offene Kapazitätsreservierung	-	-	150
Preis auf Abruf	\$	\$\$	\$\$\$
Bereitgestellte Instances	-	-	100

Offene Kapazität sreservierung verwendet	-	-	100
Verfügbare offene Kapazitätsreservie rung	-	-	50

Beispiel 2: Der Instance-Pool mit verfügbaren Reservierungen für offene Kapazitäten in der Startanfrage verfügt nicht über genügend Kapazität für Core-Knoten

In diesem Fall greift Amazon EMR auf die Einführung von Core-Knoten zurück und verwendet dabei die niedrigste Preisstrategie, wobei Kapazitätsreservierungen bestmöglich genutzt werden.

On-Demand-Strategie	Niedrigster Preis		
Angeforderte Kapazität	100		
Nutzungsstrategie	use-capacity-reservations-first		
Instance-Typ	c5.xlarge	m5.xlarge	r5.xlarge
Verfügbare offene Kapazitätsreservie rung	10	50	50
Preis auf Abruf	\$	\$\$	\$\$\$
Bereitgestellte Instances	100	-	-
Offene Kapazität sreservierung verwendet	10	-	-

Verfügbare Reservierungen für offene Kapazitäten	-	50	50

Nachdem die Instance-Flotte gestartet wurde, können Sie [describe-capacity-reservations](#) ausführen, um zu sehen, wie viele ungenutzte Kapazitätsreservierungen verbleiben.

Konfigurieren Sie Instance-Flotten so, dass sie zuerst offene Kapazitätsreservierungen verwenden

Wenn Sie die RunJobFlow-Aktion verwenden, um einen auf Instance-Flotten basierenden Cluster zu erstellen, legen Sie für die On-Demand-Zuweisungsstrategie die Optionen `lowest-price` und `UsageStrategy` für `CapacityReservationOptions` auf `use-capacity-reservations-first` fest.

```
"LaunchSpecifications":
  {"OnDemandSpecification": {
    "AllocationStrategy": "lowest-price",
    "CapacityReservationOptions":
      {
        "UsageStrategy": "use-capacity-reservations-first"
      }
  }
}
```

Sie können auch die Amazon-EMR-CLI verwenden, um einen auf Instance-Flotten basierenden Cluster zu erstellen, indem Sie zunächst Kapazitätsreservierungen verwenden.

```
aws emr create-cluster \
  --name 'use-CR-first-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

  InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarge}'] \
  \

  InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarge}',
  {InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
```

```
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-first}}'}
```

Wobei gilt,

- `use-CR-first-cluster` wird durch den Namen des Clusters ersetzt, der offenen Kapazitätsreservierungen verwendet.
- `subnet-22XXXX01` wird durch die Subnetz-ID ersetzt.

Zuerst gezielte Kapazitätsreservierungen verwenden

Wenn Sie einen Amazon-EMR-Cluster bereitstellen, können Sie sich dafür entscheiden, die Zuweisungsstrategie mit dem niedrigsten Preis außer Kraft zu setzen und zuerst die verfügbaren gezielten Kapazitätsreservierungen zu verwenden. In diesem Fall bewertet Amazon EMR alle Instance-Pools mit gezielten Kapazitätsreservierungen, die in der Startanfrage angegeben wurden, und wählt den Pool mit dem niedrigsten Preis aus, der über ausreichend Kapazität verfügt, um alle angeforderten Core-Knoten zu starten. Wenn keiner der Instance-Pools mit gezielten Kapazitätsreservierungen über ausreichende Kapazitäten für Core-Knoten verfügt, fällt Amazon EMR auf den zuvor beschriebenen Fall der besten Bemühung zurück. Das heißt, Amazon EMR bewertet alle in der Startanforderung angegebenen Instance-Pools neu und wählt den Pool mit dem niedrigsten Preis aus, der über ausreichende Kapazität verfügt, um alle angeforderten Core-Knoten zu starten. Verfügbare offene Kapazitätsreservierungen, die dem Instance-Pool entsprechen, werden automatisch übernommen. Gezielte Kapazitätsreservierungen bleiben jedoch ungenutzt.

Sobald die Core-Knoten bereitgestellt sind, wird die Availability Zone ausgewählt und repariert. Amazon EMR stellt Aufgabenknoten in Instance-Pools mit gezielten Kapazitätsreservierungen bereit, beginnend mit den günstigsten zuerst, in der ausgewählten Availability Zone, bis alle Aufgabenknoten bereitgestellt sind. Amazon EMR versucht zunächst, die verfügbaren gezielten Kapazitätsreservierungen zu verwenden, die für jeden Instance-Pool in der ausgewählten Availability Zone verfügbar sind. Nur bei Bedarf verwendet Amazon EMR dann die Strategie mit dem niedrigsten Preis, um alle verbleibenden Aufgabenknoten bereitzustellen.

Im Folgenden finden Sie Anwendungsfälle der Amazon-EMR-Kapazitätzuweisungslogik, bei der zunächst gezielte Kapazitätsreservierungen verwendet werden.

Beispiel 1: Der Instance-Pool mit verfügbaren gezielten Kapazitätsreservierungen in der Startanfrage verfügt über ausreichend Kapazität für Core-Knoten

In diesem Fall startet Amazon EMR Kapazität im Instance-Pool mit verfügbaren gezielten Kapazitätsreservierungen, unabhängig vom Preis des Instance-Pools. Daher werden Ihre gezielten Kapazitätsreservierungen wann immer möglich genutzt, bis alle Core-Knoten bereitgestellt sind.

On-Demand-Strategie	Niedrigster Preis		
Nutzungsstrategie	use-capacity-reservations-first		
Angeforderte Kapazität	100		
Instance-Typ	c5.xlarge	m5.xlarge	r5.xlarge
Verfügbare gezielte Kapazitätsreservierungen	-	-	150
Preis auf Abruf	\$	\$\$	\$\$\$
Bereitgestellte Instances	-	-	100
Gezielte Kapazität sreservierung genutzt	-	-	100
Verfügbare gezielte Kapazitätsreservierungen	-	-	50

Example Beispiel 2: Der Instance-Pool mit verfügbaren gezielten Kapazitätsreservierungen in der Startanfrage verfügt nicht über ausreichende Kapazität für Core-Knoten

On-Demand-Strategie	Niedrigster Preis
Angeforderte Kapazität	100
Nutzungsstrategie	use-capacity-reservations-first

Instance-Typ	c5.xlarge	m5.xlarge	r5.xlarge
Verfügbare gezielte Kapazitätsreservierungen	10	50	50
Preis auf Abruf	\$	\$\$	\$\$\$
Bereitgestellte Instances	100	-	-
Gezielte Kapazitätssreservierungen verwendet	10	-	-
Verfügbare gezielte Kapazitätsreservierungen	-	50	50

Nachdem die Instance-Flotte gestartet wurde, können Sie [describe-capacity-reservations](#) ausführen, um zu sehen, wie viele ungenutzte Kapazitätsreservierungen verbleiben.

Instance-Flotten so konfigurieren, dass sie zuerst gezielte Kapazitätsreservierungen verwenden

Wenn Sie die RunJobFlow-Aktion verwenden, um einen auf Instance-Flotten basierenden Cluster zu erstellen, legen Sie für die On-Demand-Zuweisungsstrategie die Optionen `lowest-price` und `UsageStrategy` für `CapacityReservationOptions` auf `use-capacity-reservations-first` und `CapacityReservationResourceGroupArn` für `CapacityReservationOptions` auf `<your resource group ARN>` fest. Weitere Informationen finden Sie unter [Arbeiten mit Kapazitätsreservierungen](#) im EC2 Amazon-Benutzerhandbuch.

```
"LaunchSpecifications":
  {"OnDemandSpecification": {
    "AllocationStrategy": "lowest-price",
    "CapacityReservationOptions":
      {
        "UsageStrategy": "use-capacity-reservations-first",
        "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup"
      }
  }
```

```
}
}
```

Wobei `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` durch Ihren Ressourcengruppen-ARN ersetzt wird.

Sie können auch die Amazon-EMR-CLI verwenden, um mithilfe gezielter Kapazitätsreservierungen einen auf Instance-Flotten basierenden Cluster zu erstellen.

```
aws emr create-cluster \
  --name 'targeted-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets
  InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarge}'] \
  InstanceFleetType=CORE,TargetOnDemandCapacity=100,\
  InstanceTypeConfigs=['{InstanceType=c5.xlarge},{InstanceType=m5.xlarge},\
  {InstanceType=r5.xlarge}'],\
  LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
  price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-
  first,CapacityReservationResourceGroupArn=arn:aws:resource-groups:sa-
  east-1:123456789012:group/MyCRGroup}}' }
```

Wobei gilt,

- `targeted-CR-cluster` wird mithilfe von gezielten Kapazitätsreservierungen durch den Namen Ihres Clusters ersetzt.
- `subnet-22XXXX01` wird durch die Subnetz-ID ersetzt.
- `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` wird durch Ihren Ressourcengruppe-ARN ersetzt.

Vermeiden Sie es, verfügbare offene Kapazitätsreservierungen zu verwenden

Example

Wenn Sie vermeiden möchten, dass Ihre offenen Kapazitätsreservierungen beim Start eines Amazon-EMR-Clusters unerwartet in Anspruch genommen werden, legen Sie die On-Demand-Zuweisungsstrategie auf `lowest-price` und `CapacityReservationPreference`

für `CapacityReservationOptions` auf `none`. Andernfalls setzt Amazon EMR die Kapazitätsreservierungspräferenz der On-Demand-Instance standardmäßig auf `open` und versucht, verfügbare offene Kapazitätsreservierungen nach bestem Wissen zu verwenden.

```
"LaunchSpecifications":
  {"OnDemandSpecification": {
    "AllocationStrategy": "lowest-price",
    "CapacityReservationOptions":
      {
        "CapacityReservationPreference": "none"
      }
  }
}
```

Sie können auch die Amazon-EMR-CLI verwenden, um einen auf einer Instance-Flotte basierenden Cluster zu erstellen, ohne offene Kapazitätsreservierungen zu verwenden.

```
aws emr create-cluster \
  --name 'none-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

  InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarge}'] \
  \

  InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarge}',
  '{InstanceType=m5.xlarge}', '{InstanceType=r5.xlarge}'], \
  LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
  price,CapacityReservationOptions={CapacityReservationPreference=none}}'}
```

Wobei gilt,

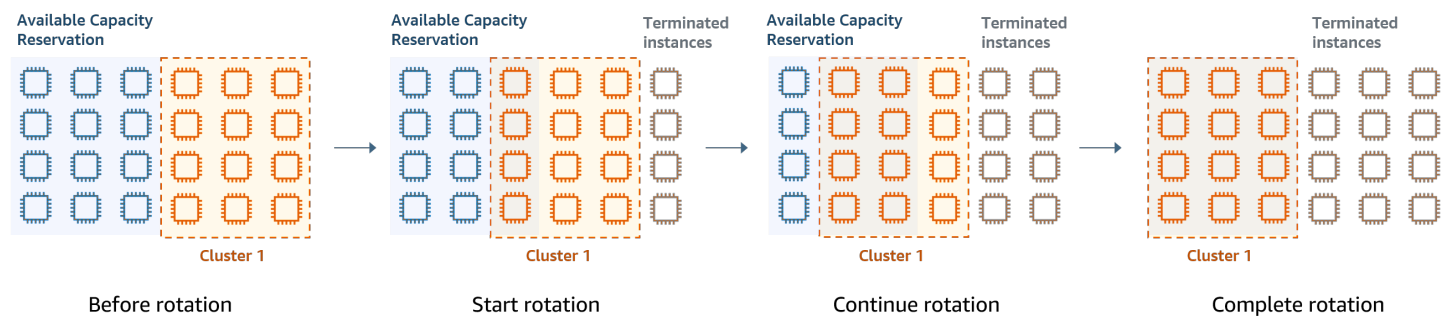
- `none-CR-cluster` wird durch den Namen Ihres Clusters ersetzt, der keine offenen Kapazitätsreservierungen verwendet.
- `subnet-22XXXX01` wird durch die Subnetz-ID ersetzt.

Szenarien für die Verwendung von Kapazitätsreservierungen

In den folgenden Szenarien können Sie von der Verwendung von Kapazitätsreservierungen profitieren.

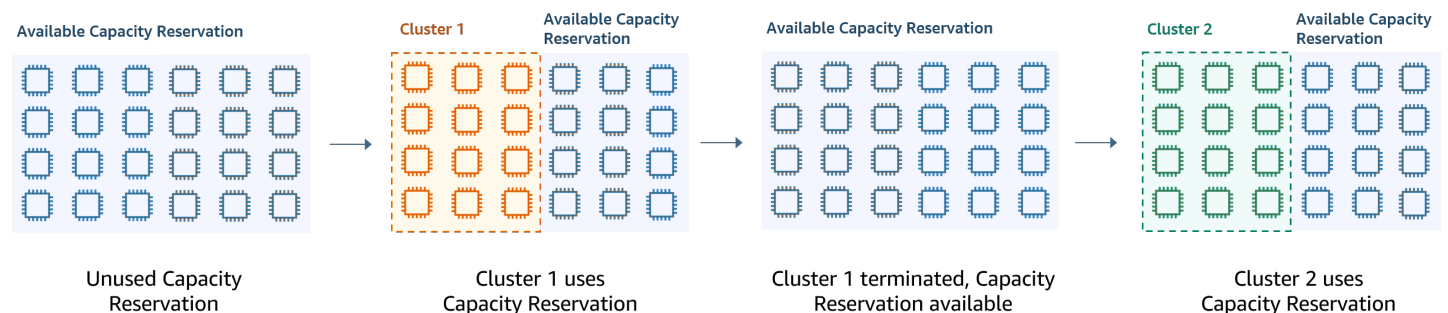
Szenario 1: Rotation eines Clusters mit langer Laufzeit mithilfe von Kapazitätsreservierungen

Wenn Sie einen Cluster mit langer Laufzeit rotieren, stellen Sie möglicherweise strenge Anforderungen an die Instance-Typen und Availability Zones für die neuen Instances, die Sie bereitstellen. Mit Kapazitätsreservierungen können Sie die Kapazitätssicherung verwenden, um die Cluster-Rotation ohne Unterbrechungen abzuschließen.



Szenario 2: Bereitstellung aufeinanderfolgender kurzlebiger Cluster mithilfe von Kapazitätsreservierungen

Sie können Kapazitätsreservierungen auch verwenden, um eine Gruppe aufeinanderfolgender, kurzlebiger Cluster für einzelne Workloads bereitzustellen, sodass, wenn Sie einen Cluster beenden, der nächste Cluster die Kapazitätsreservierungen nutzen kann. Sie können gezielte Kapazitätsreservierungen verwenden, um sicherzustellen, dass nur die vorgesehenen Cluster die Kapazitätsreservierungen nutzen.



Konfigurieren Sie einheitliche Instanzgruppen für Ihren Amazon EMR-Cluster

Mit der Instance-Gruppenkonfiguration besteht jeder Knotentyp (Master-, Core- oder Aufgabenknoten) aus demselben Instance-Typ und derselben Kaufoption für Instances: On-Demand

oder Spot. Sie geben diese Einstellungen beim Erstellen einer Instance-Gruppe an. Sie können später nicht mehr geändert werden. Sie können Core- und Aufgaben-Instance-Gruppen jedoch Instances desselben Typs und derselben Kaufoption hinzufügen. Außerdem können Sie Instances entfernen.

Wenn die On-Demand-Instances des Clusters den in Ihrem Konto verfügbaren Attributen der offenen Kapazitätsreservierungen (Instance-Typ, Plattform, Tenancy und Availability Zone) entsprechen, werden die Kapazitätsreservierungen automatisch angewendet. Sie können offene Kapazitätsreservierungen für Primär-, Kern- und Aufgabenknoten verwenden. Sie können jedoch keine gezielten Kapazitätsreservierungen verwenden oder verhindern, dass Instances offene Kapazitätsreservierungen mit übereinstimmenden Attributen starten, wenn Sie Cluster mithilfe von Instance-Gruppen bereitstellen. Wenn Sie gezielte Kapazitätsreservierungen verwenden oder verhindern möchten, dass Instances aufgrund offener Kapazitätsreservierungen starten, verwenden Sie stattdessen Instance-Flotten. Weitere Informationen finden Sie unter [Verwenden Sie Kapazitätsreservierungen mit Instance-Flotten in Amazon EMR](#).

Zum Hinzufügen verschiedener Instance-Typen nach dem Erstellen eines Clusters können Sie zusätzliche Aufgaben-Instance-Gruppen hinzufügen. Sie können verschiedene Instance-Typen und Kaufoptionen für jede Instance-Gruppe auswählen. Weitere Informationen finden Sie unter [Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen](#).

Beim Starten von Instances ist die Kapazitätsreservierungspräferenz der On-Demand-Instance standardmäßig auf open gesetzt, wodurch sie in jeder offenen Kapazitätsreservierung ausgeführt werden kann, die über passende Attribute (Instance-Typ, Plattform, Verfügbarkeitszone) verfügt. Weitere Informationen über On-Demand-Kapazitätsreservierungen finden Sie unter [Verwenden Sie Kapazitätsreservierungen mit Instance-Flotten in Amazon EMR](#).

In diesem Abschnitt wird das Erstellen eines Clusters mit einheitlichen Instance-Gruppen beschrieben. Weitere Informationen zum Ändern einer vorhandenen Instance-Gruppe durch Hinzufügen oder Entfernen von Instances manuell oder automatisch mit Auto Scaling finden Sie unter [Amazon EMR-Cluster verwalten](#).

Die Konsole zum Konfigurieren einheitlicher Instance-Gruppen verwenden

Console

So erstellen Sie einen Cluster mit Instance-Gruppen mithilfe der neuen Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Clusterkonfiguration die Option Instance-Gruppen aus.
4. Unter Knotengruppen gibt es einen Abschnitt für jeden Knotengruppentyp. Aktivieren Sie für die Primärknotengruppe das Kontrollkästchen Mehrere Primärknoten verwenden, wenn Sie drei Primärknoten haben möchten. Aktivieren Sie das Kontrollkästchen Spot-Kaufoption verwenden, wenn Sie Spot-Kauf verwenden möchten.
5. Wählen Sie für die Primär- und Core-Knotengruppen die Option Instance-Typ hinzufügen und wählen Sie bis zu 5 Instance-Typen aus. Wählen Sie für die Aufgabengruppe Instance-Typ hinzufügen und wählen Sie bis zu fünfzehn Instance-Typen aus. Amazon EMR kann jede beliebige Kombination dieser Instance-Typen beim Starten des Clusters bereitstellen.
6. Wählen Sie unter jedem Knotengruppentyp das Drop-Down-Menü Aktionen neben jeder Instance aus, um diese Einstellungen zu ändern:

EBS-Volumes hinzufügen

Geben Sie EBS-Volumes an, die an den Instance-Typ angefügt werden sollen, nachdem Amazon EMR ihn bereitgestellt hat.

Den maximalen Spot-Preis bearbeiten

Geben Sie für jeden Instance-Typ in einer Flotte einen maximalen Spot-Preis an. Sie können den Preis entweder als Prozentsatz des On-Demand-Preises oder als einen bestimmten Betrag in US-Dollar festlegen. Amazon EMR stellt Spot Instances bereit, wenn der aktuelle Spot-Preis in einer Availability Zone unter Ihrem maximalen Spot-Preis liegt. Sie zahlen den Spot-Preis, nicht unbedingt den maximalen Spot-Preis.

7. Erweitern Sie optional die Knotenkonfiguration, um eine JSON-Konfiguration einzugeben oder JSON aus Amazon S3 zu laden.
8. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
9. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

Verwenden Sie den AWS CLI , um einen Cluster mit einheitlichen Instanzgruppen zu erstellen

Um die Instance-Gruppenkonfiguration für einen Cluster mithilfe der AWS CLI anzugeben, verwenden Sie den Befehl `create-cluster` zusammen mit dem Parameter `--instance-groups`. Amazon EMR geht von der On-Demand-Instance-Option aus, es sei denn, Sie legen das Argument `BidPrice` für eine Instance-Gruppe fest. Beispiele der Befehle `create-cluster`, mit denen einheitliche Instance-Gruppen mit On-Demand-Instances gestartet werden, und eine Vielzahl von Cluster-Optionen sehen Sie, wenn Sie `aws emr create-cluster help` in der Befehlszeile eingeben oder den Abschnitt [create-cluster](#) in der AWS CLI -Befehlsreferenz lesen.

Sie können die verwenden AWS CLI , um einheitliche Instanzgruppen in einem Cluster zu erstellen, die Spot-Instances verwenden. Der angebotene Spot-Preis hängt der von Availability Zone ab. Wenn Sie die CLI oder API verwenden, können Sie die Availability Zone entweder mit dem `AvailabilityZone` Argument (wenn Sie ein EC2 -classic-Netzwerk verwenden) oder dem `SubnetID` Argument des `--ec2-attributes` Parameters angeben. Die ausgewählte Availability Zone oder das Subnetz gilt für den Cluster und wird daher für alle Instance-Gruppen verwendet. Wenn Sie keine Availability Zone oder kein Subnetz explizit angeben, wählt Amazon EMR beim Starten des Clusters die Availability Zone mit dem niedrigsten Spot-Preis aus.

Das folgende Beispiel zeigt einen Befehl `create-cluster`, mit dem Primär-, Core- und zwei Aufgaben-Instance-Gruppen erstellt werden, die alle Spot Instances verwenden. *myKey* Ersetzen Sie es durch den Namen Ihres EC2 Amazon-Schlüsselpaars.

Note

Linux-Zeilenumbruchzeichen (`\`) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (`^`).

```
aws emr create-cluster --name "MySpotCluster" \  
  --release-label emr-7.9.0 \  
  --use-default-roles \  
  --ec2-attributes KeyName=myKey \  
  --instance-groups \  
    InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,BidPrice=0.25 \  
    InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.03 \  
    InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=4,BidPrice=0.03 \  
    InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.04
```

Mithilfe der CLI können Sie einheitliche Instance-Gruppen-Cluster erstellen, die für jeden Instance-Typ in der Instance-Gruppe ein eindeutiges benutzerdefiniertes AMI angeben. Auf diese Weise können Sie verschiedene Instance-Architekturen in derselben Instance-Gruppe verwenden. Jeder Instance-Typ muss ein benutzerdefiniertes AMI mit einer passenden Architektur verwenden. Sie würden beispielsweise einen m5.xlarge-Instance-Typ mit einem benutzerdefinierten X86_64-Architektur-AMI und einen m6g.xlarge-Instance-Typ mit einem entsprechenden benutzerdefinierten AWS AARCH64 (ARM)-Architektur-AMI konfigurieren.

Das folgende Beispiel zeigt einen einheitlichen Instance-Gruppen-Cluster, der mit zwei Instance-Typen erstellt wurde, von denen jeder sein eigenes benutzerdefiniertes AMI hat. Beachten Sie, dass die benutzerdefinierten AMIs Werte nur auf Instance-Typ-Ebene angegeben werden, nicht auf Cluster-Ebene. Dadurch sollen Konflikte zwischen dem Instance-Typ AMIs und einem AMI auf Cluster-Ebene vermieden werden, die dazu führen würden, dass der Clusterstart fehlschlägt.

```
aws emr create-cluster
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-groups \

  InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
  \

  InstanceGroupType=CORE,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-234567
```

Sie können einer Instanzgruppe AMIs, die Sie einem laufenden Cluster hinzufügen, mehrere benutzerdefinierte Instanzen hinzufügen. Das CustomAmiId-Argument kann zusammen mit dem add-instance-groups-Befehl verwendet werden, wie im folgenden Beispiel gezeigt.

```
aws emr add-instance-groups --cluster-id j-123456 \
  --instance-groups \

  InstanceGroupType=Task,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
```

Verwenden des Java-SDKs zum Erstellen einer Instance-Gruppe

Instanzieren Sie ein Objekt InstanceGroupConfig, das die Konfiguration einer Instance-Gruppe für einen Cluster angibt. Um Spot-Instances zu verwenden, legen Sie die Eigenschaften withBidPrice und withMarket für das Objekt InstanceGroupConfig fest. Der folgende Code

zeigt, wie Primär-, Core- und Aufgaben-Instance-Gruppen definiert werden, die Spot Instances ausführen.

```
InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
    .withInstanceCount(1)
    .withInstanceRole("MASTER")
    .withInstanceType("m4.large")
    .withMarket("SPOT")
    .withBidPrice("0.25");

InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
    .withInstanceCount(4)
    .withInstanceRole("CORE")
    .withInstanceType("m4.large")
    .withMarket("SPOT")
    .withBidPrice("0.03");

InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
    .withInstanceCount(2)
    .withInstanceRole("TASK")
    .withInstanceType("m4.large")
    .withMarket("SPOT")
    .withBidPrice("0.10");
```

Flexibilität der Availability Zone für einen Amazon EMR-Cluster

Jede AWS-Region hat mehrere isolierte Standorte, die als Availability Zones bezeichnet werden. Beim Starten einer Instance können Sie optional eine Availability Zone (AZ) oder AWS-Region in der Region angeben, die Sie verwenden. Die [Flexibilität der Availability Zone](#) ist die Verteilung von Instanzen auf mehrere AZs. Wenn eine Instance ausfällt, können Sie Ihre Anwendung so gestalten, dass eine Instance in einer anderen AZ Anfragen verarbeiten kann. Weitere Informationen zu Availability Zones finden Sie in der Dokumentation zu [Regionen und Zonen](#) im EC2 Amazon-Benutzerhandbuch.

[Instance-Flexibilität](#) ist die Verwendung mehrerer Instance-Typen zur Erfüllung der Kapazitätsanforderungen. Wenn Sie Flexibilität bei Instances zum Ausdruck bringen, können Sie die Gesamtkapazität für alle Instance-Größen, Familien und Generationen nutzen. Im Vergleich zu einem Cluster, der einen einzigen Instance-Typ verwendet, verbessert sich die Wahrscheinlichkeit, die erforderliche Menge an Rechenkapazität zu finden und zuzuweisen.

Durch die Flexibilität von Instances und Availability Zones werden [Fehler bei unzureichender Kapazität \(ICE\)](#) und Spot-Unterbrechungen im Vergleich zu einem Cluster mit einem einzigen Instance-Typ oder AZ reduziert. Verwenden Sie die hier beschriebenen bewährten Methoden, um zu bestimmen, welche Instances Sie diversifizieren sollten, nachdem Sie die ursprüngliche Instance-Familie und Größe kennen. Dieser Ansatz maximiert die Verfügbarkeit von EC2 Amazon-Kapazitätspools bei minimalen Leistungs- und Kostenabweichungen.

Flexibilität in Bezug auf Availability Zones

Wir empfehlen, dass Sie alle Availability Zones für die Verwendung in Ihrer Virtual Private Cloud (VPC) konfigurieren und sie für Ihren EMR-Cluster auswählen. Cluster dürfen nur in einer Availability Zone existieren, aber mit Amazon-EMR-Instance-Flotten können Sie mehrere Subnetze für verschiedene Availability Zones auswählen. Wenn Amazon EMR den Cluster startet, sieht er auf diese Subnetze, um die Instances und Kaufoptionen zu finden, die Sie angeben. Wenn Sie einen EMR-Cluster für mehrere Subnetze bereitstellen, kann Ihr Cluster im Vergleich zu Clustern in einem einzelnen Subnetz auf einen größeren EC2 Amazon-Kapazitätspool zugreifen.

Wenn Sie eine bestimmte Anzahl von Availability Zones für die Verwendung in Ihrer Virtual Private Cloud (VPC) für Ihren EMR-Cluster priorisieren müssen, können Sie die Spot Placement Score-Funktion bei Amazon nutzen. EC2 Mit der Spot-Platzierungsbewertung geben Sie die Rechenanforderungen für Ihre Spot-Instances an und geben dann die zehn besten AWS-Regionen oder Availability Zones EC2 zurück, die auf einer Skala von 1 bis 10 bewertet wurden. Eine Punktzahl von 10 zeigt an, dass Ihre Spot-Anforderung sehr wahrscheinlich erfolgreich sein wird. Eine Punktzahl von 1 zeigt an, dass Ihre Spot-Anforderung sehr wahrscheinlich nicht erfolgreich sein wird. Weitere Informationen zur Verwendung von Spot Placement Scoring finden Sie unter [Spot Placement Score](#) im EC2 Amazon-Benutzerhandbuch.

Flexibel sein bei Instance-Typen

Instance-Flexibilität ist die Verwendung mehrerer Instance-Typen zur Erfüllung der Kapazitätsanforderungen. Die Instance-Flexibilität kommt sowohl der Nutzung von Amazon EC2 Spot als auch der On-Demand-Instance zugute. Dank der Instance-Flexibilität von Spot-Instances kann Amazon EC2 Instances mithilfe von Echtzeit-Kapazitätsdaten aus tieferen Kapazitätspools starten. Außerdem wird vorhergesagt, welche Instances am verfügbarsten sind. Dies bietet weniger Unterbrechungen und kann die Gesamtkosten eines Workloads reduzieren. Mit On-Demand-Instances reduziert die Instance-Flexibilität Fehler bei unzureichender Kapazität (ICE), wenn die Gesamtkapazität über eine größere Anzahl von Instance-Pools bereitgestellt wird.

Für Instance-Gruppen-Cluster können Sie bis zu 50 EC2 Instance-Typen angeben. Für Instanzflotten mit Zuweisungsstrategie können Sie bis zu 30 EC2 Instanztypen für jede Primär-, Kern- und Taskknotengruppe angeben. Eine breitere Palette von Instances verbessert die Vorteile der Instance-Flexibilität.

Ausdrücken der Instance-Flexibilität

Beachten Sie die folgenden bewährten Methoden, um die Instance-Flexibilität für Ihre Anwendung zum Ausdruck zu bringen.

Themen

- [Die Instance-Familie und -größe ermitteln](#)
- [Zusätzliche Instances hinzufügen](#)

Die Instance-Familie und -größe ermitteln

Amazon EMR unterstützt verschiedene Instance-Typen für unterschiedliche Anwendungsfälle. Diese Instance-Typen sind in der [Unterstützte Instance-Typen mit Amazon EMR](#)-Dokumentation aufgeführt. Jeder Instance-Typ gehört zu einer Instance-Familie, die beschreibt, für welche Anwendung der Typ optimiert ist.

Bei neuen Workloads sollten Sie einen Vergleich mit Instance-Typen aus der Allzweckfamilie durchführen, z. B. m5 oder c5. Überwachen Sie anschließend die OS- und YARN-Metriken von Ganglia und Amazon CloudWatch zur Ermittlung von Systemengpässen bei Spitzenlast. Zu den Engpässen gehören CPU, Arbeitsspeicher, Speicher und I/O Betrieb. Nachdem Sie die Engpässe identifiziert haben, wählen Sie rechenoptimiert, arbeitsspeicheroptimiert, speicheroptimiert oder eine andere geeignete Instance-Familie für Ihre Instance-Typen. Weitere Informationen finden Sie auf der Seite [Ermitteln Sie die richtige Infrastruktur für Ihre Spark-Workloads](#) im Amazon EMR-Best-Practices-Leitfaden unter. GitHub

Identifizieren Sie als Nächstes den kleinsten YARN-Container oder Spark-Executor, den Ihre Anwendung benötigt. Dies ist die kleinste Instance-Größe, die zum Container passt, und die minimale Instance-Größe für den Cluster. Verwenden Sie diese Metrik, um Instances zu ermitteln, mit denen Sie weiter diversifizieren können. Eine kleinere Instance ermöglicht mehr Instance-Flexibilität.

Für maximale Instance-Flexibilität sollten Sie so viele Instances wie möglich nutzen. Wir empfehlen Ihnen, mit Instances zu diversifizieren, die ähnliche Hardwarespezifikationen haben. Dadurch wird der Zugriff auf EC2 Kapazitätspools bei minimalen Kosten- und Leistungsschwankungen maximiert. Diversifizieren Sie zwischen verschiedenen Größen. Priorisieren Sie dazu zuerst AWS Graviton und

frühere Generationen. Als allgemeine Regel gilt: Versuchen Sie, für jeden Workload flexibel über mindestens 15 Instance-Typen hinweg zu sein. Wir empfehlen, mit allgemeinen, rechenoptimierten oder arbeitsspeicheroptimierten Instances zu beginnen. Diese Instance-Typen bieten die größte Flexibilität.

Zusätzliche Instances hinzufügen

Fügen Sie für eine maximale Vielfalt zusätzliche Instance-Typen hinzu. Priorisieren Sie zuerst die Instance-Größe, Graviton und Generierungsflexibilität. Dies ermöglicht den Zugriff auf zusätzliche EC2 Kapazitätspools mit ähnlichen Kosten- und Leistungsprofilen. Wenn Sie aufgrund von ICE- oder punktuellen Unterbrechungen mehr Flexibilität benötigen, sollten Sie die Flexibilität von Varianten und Produktfamilien in Betracht ziehen. Jeder Ansatz hat Kompromisse, die von Ihrem Anwendungsfall und Ihren Anforderungen abhängen.

- **Größenflexibilität** – Diversifizieren Sie zunächst mit Instances unterschiedlicher Größe innerhalb derselben Produktfamilie. Instances innerhalb derselben Familie bieten dieselben Kosten und dieselbe Leistung, können aber auf jedem Host eine unterschiedliche Anzahl von Containern starten. Wenn die Mindestgröße des Executors, die Sie benötigen, 2 vCPU und 8 GB Arbeitsspeicher beträgt, beträgt die Mindestgröße der Instance `m5.xlarge`. Geben Sie aus Gründen der Größenflexibilität `m5.xlarge`, `m5.2xlarge`, `m5.4xlarge`, `m5.8xlarge`, `m5.12xlarge`, `m5.16xlarge` und `m5.24xlarge` an.
- **Graviton-Flexibilität** – Neben der Größe können Sie mit Graviton-Instances auch eine größere Vielfalt an Optionen erzielen. Graviton-Instances werden von AWS Graviton2-Prozessoren angetrieben, die das beste Preis-Leistungs-Verhältnis für Cloud-Workloads bei Amazon bieten. EC2 Mit der minimalen Instance-Größe von `m5.xlarge` können Sie beispielsweise `m6g.xlarge`, `m6g.2xlarge`, `m6g.4xlarge`, `m6g.8xlarge` und `m6g.16xlarge` für die Graviton-Flexibilität einschließen.
- **Flexibilität bei der Generierung** – Ähnlich wie Graviton und Größenflexibilität haben auch Instances der Familien früherer Generationen dieselben Hardwarespezifikationen. Dies führt zu einem ähnlichen Kosten- und Leistungsprofil mit einer Erhöhung des insgesamt zugänglichen EC2 Amazon-Pools. Für Flexibilität bei der Generierung schließen Sie `m4.xlarge`, `m4.2xlarge`, `m4.10xlarge` und `m4.16xlarge` ein.
- **Familien- und Variantenflexibilität**
 - **Kapazität** – Um die Kapazität zu optimieren, empfehlen wir Instance-Flexibilität für alle Instance-Familien. Gängige Instances aus verschiedenen Instance-Familien verfügen über tiefere Instance-Pools, die bei der Erfüllung der Kapazitätsanforderungen helfen können. Instances aus verschiedenen Familien haben jedoch unterschiedliche Verhältnisse zwischen vCPU und

Arbeitsspeicher. Dies führt zu einer Unterauslastung, wenn der erwartete Anwendungscontainer für eine andere Instance dimensioniert ist. Schließen Sie beispielsweise mit `m5.xlarge` für Datenverarbeitung optimierte Instances wie `c5` oder arbeitsspeicheroptimierte Instances wie `r5` ein, um die Flexibilität der Instance-Familie zu gewährleisten.

- **Kosten** – Zur Kostenoptimierung empfehlen wir die variantenübergreifende Instance-Flexibilität. Diese Instances haben das gleiche Speicher- und vCPU-Verhältnis wie die ursprüngliche Instance. Der Nachteil bei der Variantenflexibilität besteht darin, dass diese Instances kleinere Kapazitätspools haben, was zu begrenzter zusätzlicher Kapazität oder höheren Spot-Unterbrechungen führen kann. Zu `m5.xlarge` zählen beispielsweise AMD-basierte Instances (`m5a`), SSD-basierte Instances (`m5d`) oder netzwerkoptimierte Instances (`m5n`) für beispielsweise Variantenflexibilität.

Konfiguration von Amazon EMR-Cluster-Instance-Typen und Best Practices für Spot-Instances

Verwenden Sie die Richtlinien in diesem Abschnitt zum Festlegen der Instance-Typen, Kaufoptionen und Speicherkapazität, die für jeden Knotentyp in einem EMR-Cluster bereitgestellt wird.

Welchen Instance-Typ sollten Sie verwenden?

Es gibt mehrere Möglichkeiten, EC2 Amazon-Instances zu einem Cluster hinzuzufügen. Welche Methode Sie wählen sollten, hängt davon ab, ob Sie die Instance-Gruppen-Konfiguration oder die Instance-Flotten-Konfiguration für den Cluster verwenden.

- **Instance-Gruppen**
 - Fügen Sie vorhandenen Core- und Task-Instance-Gruppen manuell Instances desselben Typs hinzu.
 - Fügen Sie manuell eine Task-Instance-Gruppe hinzu, die einen anderen Instance-Typ verwenden kann.
 - Richten Sie die automatische Skalierung in Amazon EMR für eine Instance-Gruppe ein und fügen Sie Instances automatisch hinzu und entfernen Sie sie basierend auf dem Wert einer von Ihnen angegebenen CloudWatch Amazon-Metrik. Weitere Informationen finden Sie unter [Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen](#).
- **Instance-Flotten**
 - Fügen Sie eine einzelne Task-Instance-Flotte hinzu.

- Ändern Sie die Zielkapazität für On-Demand- und Spot-Instances für vorhandene Core- und Task-Instance-Flotten. Weitere Informationen finden Sie unter [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#).

Eine Möglichkeit zum Planen der Instances Ihres Clusters ist die Ausführung eines Test-Clusters mit einem repräsentativen Beispielsatz von Daten und die Überwachung der Auslastung der Knoten im Cluster. Weitere Informationen finden Sie unter [Einen Amazon EMR-Cluster bei der Ausführung seiner Arbeit anzeigen und überwachen](#). Eine andere Möglichkeit besteht in der Berechnung der Kapazität der Instances, die Sie erwägen, und im Vergleichen dieses Werts mit der Größe Ihrer Daten.

Im Allgemeinen benötigt der primäre Knotentyp, der Aufgaben zuweist, keine EC2 Instance mit viel Rechenleistung. EC2 Amazon-Instances für den Core-Knotentyp, die Aufgaben verarbeiten und Daten in HDFS speichern, benötigen sowohl Rechenleistung als auch Speicherkapazität; EC2 Amazon-Instances für den Task-Knotentyp, die keine Daten speichern, benötigen nur Rechenleistung. Richtlinien zu verfügbaren EC2 Amazon-Instances und deren Konfiguration finden Sie unter [EC2 Amazon-Instance-Typen für die Verwendung mit Amazon EMR konfigurieren](#).

Die folgenden Richtlinien gelten für die meisten Amazon-EMR-Cluster.

- Es gibt ein vCPU-Limit für die Gesamtzahl der EC2 On-Demand-Amazon-Instances, die Sie auf einem AWS Konto pro AWS-Region ausführen. Weitere Informationen zum vCPU-Limit und dazu, wie Sie eine Limiterhöhung für Ihr Konto beantragen können, finden Sie unter [On-Demand-Instances](#) im EC2Amazon-Benutzerhandbuch für Linux-Instances.
- Der Primärknoten stellt keine großen Datenverarbeitungsanforderungen. Für Cluster mit einer großen Anzahl von Knoten oder für Cluster mit Anwendungen, die speziell auf dem primären Knoten (JupyterHub, Hue usw.) bereitgestellt werden, ist möglicherweise ein größerer primärer Knoten erforderlich, der zur Verbesserung der Cluster-Leistung beitragen kann. Erwägen Sie beispielsweise, eine m5.xlarge-Instance für kleine Cluster (50 oder weniger Knoten) zu verwenden und für größere Cluster auf einen größeren Instance-Typ umzusteigen.
- Die benötigte Rechenleistung der Core- und Aufgabenknoten hängt von der Art der Verarbeitung ab, die Ihre Anwendung durchführt. Viele Aufträge können auf Allzweck-Instance-Typen ausgeführt werden, die eine ausgewogene Leistung in Bezug auf CPU, Festplattenspeicher und I/O-Durchsatz bieten. Rechenintensive Cluster profitieren von der Ausführung auf High-CPU-Instances, die proportional über mehr CPU-Leistung verfügen als RAM. Datenbank- und Arbeitsspeicher-Caching-Anwendungen können von der Ausführung auf High-Memory-Instances profitieren. Netzwerkintensive und CPU-intensive Anwendungen wie Analyse-, NLP- und Machine Learning-

Tools, können von der Ausführung auf Cluster-Datenverarbeitungs-Instances profitieren, da diese proportional hohe CPU-Ressourcen und eine erhöhte Netzwerkleistung bieten.

- Wenn einzelne Phasen Ihres Clusters unterschiedliche Kapazitätserfordernisse haben, können Sie mit einer geringen Anzahl von Core-Knoten beginnen und die Anzahl von Aufgabenknoten den wechselnden Anforderungen der Auftragsverlaufskapazität entsprechend erhöhen oder verringern.
- Die Menge der Daten, die Sie verarbeiten können, hängt von der Kapazität Ihrer Core-Knoten und der Datenmenge als Eingabe, während der Verarbeitung, und als Ausgabe ab. Die Eingabe-, intermediären und Ausgabedatensätze befinden sich während der Verarbeitung alle auf dem Cluster.

Wann sollten Sie Spot Instances verwenden?

Wenn Sie einen Cluster in Amazon EMR starten, können Sie auswählen, ob Sie Primär-, Core- oder Aufgaben-Instances auf Spot Instances starten möchten. Da jeder Typ von Instance-Gruppe eine andere Rolle im Cluster hat, hat das Starten der einzelnen Knotentypen auf Spot-Instances bestimmte Auswirkungen. Sie können eine Instance-Kaufoption nicht ändern, während der Cluster ausgeführt wird. Um On-Demand-Instances in Spot Instances oder umgekehrt zu ändern, müssen Sie im Fall von Primär- und Core-Knoten den Cluster beenden und einen neuen Cluster starten. Im Fall von Aufgabenknoten können Sie eine neue Aufgaben-Instance-Gruppe oder -Flotte starten und die alte entfernen.

Themen

- [Amazon-EMR-Einstellungen, die Aufgabenfehler aufgrund des Beendens von Aufgabenknoten-Spot Instances verhindern](#)
- [Primärknoten auf einer Spot Instance](#)
- [Core-Knoten auf Spot Instances](#)
- [Aufgabenknoten auf Spot Instances](#)
- [Instance-Konfigurationen für Anwendungsszenarien](#)

Amazon-EMR-Einstellungen, die Aufgabenfehler aufgrund des Beendens von Aufgabenknoten-Spot Instances verhindern

Da Spot Instances häufig zum Ausführen von Aufgabenknoten verwendet werden, verfügt Amazon EMR über Standardfunktionen für die Planung von YARN-Aufträgen, sodass laufende Aufträge nicht fehlschlagen, wenn Aufgabenknoten, die auf Spot Instances ausgeführt werden, beendet werden.

Amazon EMR ermöglicht dies, indem Anwendungsmasterprozesse nur auf Core-Knoten ausgeführt werden können. Der Anwendungsmasterprozess steuert die Ausführung von Aufträgen und muss während der gesamten Laufzeit des Auftrags aktiv bleiben.

Amazon-EMR-Version 5.19.0 und höher verwendet zu diesem Zweck das integrierte [YARN-Knotenbeschriftungsfeature](#). (Frühere Versionen verwendeten einen Code-Patch). Die Eigenschaften in den Klassifizierungen `yarn-site` und in der `capacity-scheduler`-Konfiguration sind standardmäßig so konfiguriert, dass der YARN-Kapazitätsplaner und der Fair-Scheduler die Vorteile von Knotenbezeichnungen nutzen. Amazon EMR kennzeichnet Core-Knoten automatisch mit dem CORE-Label und legt Eigenschaften fest, sodass Anwendungsmaster nur für Knoten mit dem CORE-Label geplant werden. Durch manuelles Ändern verwandter Eigenschaften in den Konfigurationsklassifizierungen von Yarn-Site und Kapazitätsplaner oder direkt in den zugehörigen XML-Dateien könnte diese Feature beeinträchtigt oder verändert werden.

Amazon EMR konfiguriert die folgenden Eigenschaften und Werte standardmäßig. Seien Sie vorsichtig, wenn Sie diese Eigenschaften konfigurieren.

 Note

Beginnend mit der Amazon-EMR-6.x-Release-Reihe ist das Feature YARN-Knotenbeschriftungen standardmäßig deaktiviert. Die Anwendungs-Primär-Prozesse können standardmäßig sowohl auf Core- als auch auf Aufgabenknoten ausgeführt werden. Sie können die Funktion für YARN-Knotenbeschriftungen aktivieren, indem Sie folgende Eigenschaften konfigurieren:

- `yarn.node-labels.enabled: true`
- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

- `yarn-site` (`yarn-site.xml`) auf allen Knoten
 - `yarn.node-labels.enabled: true`
 - `yarn.node-labels.am.default-node-label-expression: 'CORE'`
 - `yarn.node-labels.fs-store.root-dir: '/apps/yarn/nodelabels'`
 - `yarn.node-labels.configuration-type: 'distributed'`
- `yarn-site` (`yarn-site.xml`) auf Primär- und Core-Knoten
 - `yarn.nodemanager.node-labels.provider: 'config'`
 - `yarn.nodemanager.node-labels.provider.configured-node-partition: 'CORE'`

- `capacity-scheduler (capacity-scheduler.xml)` auf allen Knoten
 - `yarn.scheduler.capacity.root.accessible-node-labels: '*'`
 - `yarn.scheduler.capacity.root.accessible-node-labels.CORE.capacity: 100`
 - `yarn.scheduler.capacity.root.default.accessible-node-labels: '*'`
 - `yarn.scheduler.capacity.root.default.accessible-node-labels.CORE.capacity: 100`

Primärknoten auf einer Spot Instance

Der Primärknoten kontrolliert und steuert den Cluster. Wenn er beendet wird, wird der Cluster beendet. Daher sollten Sie den Primärknoten nur als Spot Instance starten, wenn Sie einen Cluster ausführen, dessen plötzliche Beendigung akzeptabel ist. Dies kann der Fall sein, wenn Sie eine neue Anwendung testen, wenn Sie einen Cluster vorliegen haben, der Daten in regelmäßigen Abständen in einem externen Speicher wie Amazon S3 ablegt oder wenn Sie einen Cluster ausführen, bei dem die Kosten eine wichtigere Rolle spielen als der Abschluss des Clusters.

Wenn Sie die Primär-Instance-Gruppe als Spot Instance starten, wird der Cluster erst gestartet, wenn die Spot-Instance-Anforderung erfüllt ist. Diese Tatsache muss bei der Auswahl des maximalen Spot-Preises berücksichtigt werden.

Sie können einen Spot-Instance-Primärknoten nur beim Starten des Clusters hinzufügen. Primärknoten können einem aktuell ausgeführten Cluster weder hinzugefügt noch daraus entfernt werden.

Normalerweise führen Sie den Primärknoten nur als Spot Instance aus, wenn Sie den gesamten Cluster (alle Instance-Gruppen) als Spot Instances ausführen.

Core-Knoten auf Spot Instances

Core-Knoten verarbeiten Daten und speichern Informationen mit HDFS. Das Beenden einer Core-Instance birgt das Risiko eines Datenverlusts. Aus diesem Grund sollten Sie Core-Knoten auf Spot-Instances nur dann ausführen, wenn ein teilweiser Verlust von HDFS-Daten toleriert werden kann.

Beim Starten der Core-Instance-Gruppe als Spot Instances wartet Amazon EMR vor dem Starten der Instance-Gruppe, bis alle angeforderten Core-Instances bereitgestellt werden können. Mit anderen Worten, wenn Sie sechs EC2 Amazon-Instances anfordern und nur fünf zu oder unter Ihrem maximalen Spot-Preis verfügbar sind, wird die Instance-Gruppe nicht gestartet. Amazon EMR wartet weiterhin, bis alle sechs EC2 Amazon-Instances verfügbar sind oder bis Sie den Cluster beenden. Sie können die Anzahl der Spot-Instances in einer Core-Instance-Gruppe ändern, um

einem ausgeführten Cluster Kapazitäten hinzuzufügen. Weitere Informationen zum Arbeiten mit Instance-Gruppen und zur Art, wie Spot-Instances mit Instance-Flotten funktionieren, finden Sie unter [the section called “Instance-Flotten oder Instance-Gruppen konfigurieren”](#).

Aufgabenknoten auf Spot Instances

Die Aufgabenknoten verarbeiten Daten, bewahren jedoch keine persistente Daten in HDFS auf. Wenn sie beendet werden da der Spot-Preis über Ihren maximalen Spot-Preis geklettert ist, gehen keine Daten verloren und die Auswirkung auf Ihrem Cluster ist minimal.

Wenn Sie einen oder mehrere Aufgaben-Instance-Gruppen als Spot Instances starten, stellt Amazon EMR so viele Aufgabenknoten wie zu Ihrem maximalen Spot-Preis möglich bereit. Das bedeutet, dass wenn Sie eine Aufgaben-Instance-Gruppe mit sechs Knoten anfordern und nur fünf Spot Instances zu oder unter Ihrem maximalen Spot-Preis verfügbar sind, Amazon EMR die Instance-Gruppe mit fünf Knoten startet und den sechsten nach Möglichkeit später hinzufügt.

Das Starten von Aufgaben-Instance-Gruppen als Spot-Instances stellt eine strategische Möglichkeit dar, die Kapazität Ihres Clusters zu erweitern und gleichzeitig die Kosten zu minimieren. Wenn Sie Ihre Primär- und Kern-Instance-Gruppen als On-Demand-Instances starten, ist ihre Kapazität für die Ausführung des Clusters garantiert. Sie können Ihren Instance-Gruppen nach Bedarf Task-Instances hinzufügen, um ein Spitzenaufkommen an Datenverkehr zu verarbeiten oder die Datenverarbeitung zu beschleunigen.

Sie können Task-Knoten mithilfe der Konsole oder der API hinzufügen oder entfernen. AWS CLI Sie können auch zusätzliche Aufgabengruppen hinzufügen, können aber diese nach dem Erstellen nicht mehr entfernen.

Instance-Konfigurationen für Anwendungsszenarien

Die folgende Tabelle stellt eine kurze Referenz für Knotentyp-Kaufoptionen und -Konfigurationen dar, die für bestimmte Anwendungsszenarien in der Regel geeignet sind. Klicken Sie auf den Link, um weitere Informationen zu den einzelnen Szenariotypen anzuzeigen.

Anwendungsszenario	Kaufoption für Primärknoten	Kaufoption für Core-Knoten	Kaufoption für Aufgabenknoten
Langläufer-Cluster und Data Warehouses	On-Demand	On-Demand oder Instance-Flottenkombination	Spot- oder Instance-Flottenkombination

Anwendungsszenario	Kaufoption für Primärknoten	Kaufoption für Core-Knoten	Kaufoption für Aufgabenknoten
Kostengesteuerte Workloads	Spot-Instances	Spot-Instances	Spot-Instances
Datenkritische Workloads	On-Demand	On-Demand	Spot- oder Instance-Flottenkombination
Testen von Anwendungen	Spot-Instances	Spot-Instances	Spot-Instances

Es gibt verschiedene Szenarien, in denen Spot Instances für die Ausführung eines Amazon-EMR-Clusters nützlich sind.

Langläufer-Cluster und Data Warehouses

Wenn Sie einen permanenten Amazon-EMR-Cluster, z. B. ein Data Warehouse, ausführen, der vorhersehbare Schwankungen der Rechenkapazität bietet, können Sie die Nachfrage in Spitzenzeiten mit Spot Instances kostengünstiger bewältigen. Sie können Ihre Primär- und Core-Instance-Gruppen als On-Demand starten, um die normale Kapazität zu bewältigen, und die Aufgaben-Instance-Gruppe als Spot Instances für Ihre maximale Workload-Anforderungen starten.

Kostengesteuerte Workloads

Wenn Sie kurzlebige Cluster ausführen, für die niedrige Kosten wichtiger sind als die Zeit bis zum Abschluss des Vorgangs, der Verlust von Teilarbeiten akzeptabel ist, können Sie den gesamten Cluster (Primär-, Core- und Aufgaben-Instance-Gruppen) als Spot Instances ausführen, um von den größten Kosteneinsparungen zu profitieren.

Datenkritische Workloads

Wenn Sie einen Cluster ausführen, für den niedrige Kosten wichtiger sind als die Zeit bis zum Abschluss des Vorgangs, der Verlust von Teilarbeiten jedoch nicht akzeptabel ist, können Sie die Primär- und Core-Instance-Gruppen als On-Demand-Instances starten und durch eine oder mehrere Aufgaben-Instance-Gruppen der Spot Instances ergänzen. Wenn Sie die Primär- und Core-Instance-Gruppen als On-Demand-Instance ausführen, wird sichergestellt, dass Ihre Daten in HDFS persistent sind und dass der Cluster vor einer Beendigung aufgrund der Fluktuation des Spot-Markts geschützt ist. Gleichzeitig profitieren Sie von Kosteneinsparungen, die aus der Ausführung der Aufgaben-Instance-Gruppen als Spot Instances resultieren.

Testen von Anwendungen

Wenn Sie eine neue Anwendung testen, um sie für den Start in einer Produktionsumgebung vorzubereiten, können Sie den gesamten Cluster (Primär-, Core- und Aufgaben-Instance-Gruppen) als Spot Instances ausführen, um die Kosten der Tests zu senken.

Berechnen der erforderlichen HDFS-Kapazität eines Clusters

Die Größe des verfügbaren HDFS-Speicher für Ihren Cluster hängt von den folgenden Faktoren ab:

- Die Anzahl der EC2 Amazon-Instances, die für Kernknoten verwendet werden.
- Die Kapazität des EC2 Amazon-Instance-Speichers für den verwendeten Instance-Typ. Weitere Informationen zu Instance-Speicher-Volumes finden Sie unter [Amazon Amazon EC2 Instance Store](#) im EC2 Amazon-Benutzerhandbuch.
- Anzahl und Größe von Amazon-EBS-Volumes, die Core-Knoten angefügt sind.
- Ein Replikationsfaktor, der bestimmt, wie jeder Datenblock in HDFS für RAID-Redundanz gespeichert wird. Standardmäßig beträgt der Replikationsfaktor 3 für einen Cluster mit 10 oder mehr Core-Knoten, 2 für einen Cluster mit 4 bis 9 Core-Knoten und 1 für einen Cluster mit maximal 3 Knoten.

Zum Berechnen der Kapazität eines HDFS-Clusters addieren Sie für jeden Core-Knoten die Instance-Speicher-Volume-Kapazität zu der Amazon-EBS-Speicherkapazität (falls verwendet). Multiplizieren Sie das Ergebnis mit der Anzahl der Core-Knoten und dividieren Sie dann die Summe durch den Replikationsfaktor basierend auf der Anzahl der Core-Knoten. Beispiel: Ein Cluster mit 10 Core-Knoten des Typs i2.xlarge, die 800 GB Instance-Speicher ohne angeschlossenen Amazon-EBS-Volumes haben, besitzt insgesamt etwa 2 666 GB für HDFS (10 Knoten x 800 GB ÷ 3 Replikationsfaktor).

Wenn der berechnete HDFS-Kapazitätswert kleiner ist als Ihre Daten, können Sie die HDFS-Speicherkapazität wie folgt erhöhen:

- Erstellen eines Clusters mit zusätzlichen Amazon-EBS-Volumes oder Hinzufügen von Instance-Gruppen mit angefügten Amazon-EBS-Volumes zu einem vorhandenen Cluster
- Hinzufügen weiterer Core-Knoten
- Auswahl eines EC2 Amazon-Instance-Typs mit größerer Speicherkapazität
- Verwenden der Datenkomprimierung
- Ändern der Hadoop-Konfigurationseinstellungen zum Verringern des Replikationsfaktors

Verwenden Sie diese Option mit Bedacht, da durch Verringern des Replikationsfaktors die Redundanz der HDFS-Daten sowie die Cluster-Funktion zur Wiederherstellung von verlorenen oder beschädigten HDFS-Blöcken beeinträchtigt wird.

Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren

Bei der Planung Ihres Clusters müssen Sie sich unter anderem für die verfügbare Debugging-Unterstützung entscheiden. Wenn Sie Ihre Datenverarbeitungsanwendung erstmals entwickeln, empfehlen wir Ihnen, die Anwendung auf einem Cluster zu testen, indem Sie eine kleine, aber repräsentative Untermenge Ihrer Daten verarbeiten. Wenn Sie dies tun, möchten Sie wahrscheinlich die Vorteile all der Debugging-Tools in Amazon EMR nutzen, z. B. die Archivierung von Protokolldateien in Amazon S3.

Wenn Sie die Entwicklung Ihrer Anwendung abgeschlossen haben und die Datenverarbeitung in die Produktionsumgebung wechselt, können Sie das Debuggen verringern. Auf diese Weise können Sie die Kosten für die Speicherung von Protokolldateiarchiven in Amazon S3 einsparen und die Verarbeitungslast für den Cluster reduzieren, da dieser den Zustand nicht mehr zu Amazon S3 schreiben muss. Der Nachteil ist, dass Ihnen bei Problemen weniger Tools zur Verfügung stehen, um das Problem zu untersuchen.

Standardmäßige Protokolldateien

Standardmäßig schreibt jeder Cluster Protokolldateien auf dem Primärknoten. Die Dateien werden in das `/mnt/var/log/`-Verzeichnis geschrieben. Sie können darauf zugreifen, indem Sie sich, wie in [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#) beschrieben, per SSH mit dem Primärknoten verbinden. Amazon EMR sammelt bestimmte System- und Anwendungsprotokolle, die von Amazon EMR-Daemons und anderen Amazon EMR-Prozessen generiert wurden, um einen effektiven Servicebetrieb sicherzustellen.

Note

Wenn Sie Amazon-EMR-Version 6.8.0 oder früher verwenden, werden Protokolldateien während der Clusterbeendigung in Amazon S3 gespeichert, sodass Sie nicht mehr auf die Protokolldateien zugreifen können, wenn der Primärknoten beendet wird. Amazon EMR veröffentlicht 6.9.0 und höher und archiviert Protokolle während der Cluster-Herunterskalierung in Amazon S3, sodass die auf dem Cluster generierten Protokolldateien auch nach dem Beenden des Knotens bestehen bleiben.

Sie müssen nicht alles aktivieren, um die Protokolldateien auf dem Primärknoten schreiben zu lassen. Dies ist das Standardverhalten von Amazon EMR und Hadoop.

Ein Cluster generiert mehrere Arten von Protokolldateien. Diese umfassen unter anderem:

- **Schritt-Protokolle** – Diese Protokolle werden vom Amazon-EMR-Service generiert und enthalten Informationen über den Cluster und die Ergebnisse der einzelnen Schritte. Die Protokolldateien werden im `/mnt/var/log/hadoop/steps/-` Verzeichnis auf dem Primärknoten gespeichert. Jeder Schritt protokolliert seine Ergebnisse in einem separaten, nummerierten Unterverzeichnis: `/mnt/var/log/hadoop/steps/s-stepId1/` für den ersten Schritt, `/mnt/var/log/hadoop/steps/s-stepId2/` für den zweiten Schritt, und so weiter. Die 13-stellige Schritt-ID (z. B. `stepId1` `stepId2`) ist für einen Cluster eindeutig.
- **Hadoop- und YARN-Komponentenprotokolle** — Die Protokolle für Komponenten, die sowohl Apache YARN als auch zugeordnet sind MapReduce, befinden sich beispielsweise in separaten Ordnern in `/mnt/var/log`. Die Speicherorte der Protokolldateien für die Hadoop-Komponenten unter `/mnt/var/log` lauten folgendermaßen: `hadoop-hdfs`, `hadoop-mapreduce`, `hadoop-httpfs` und `hadoop-yarn`. Das `hadoop-state-pusher` Verzeichnis ist für die Ausgabe des Hadoop-State-Pusher-Prozesses vorgesehen.
- **Bootstrap-Aktion-Protokolle** – Wenn Ihr Auftrag Bootstrap-Aktionen verwendet, werden die Ergebnisse dieser Aktionen protokolliert. Die Protokolldateien werden in `/mnt/var/log/bootstrap-actions/` auf dem Primärknoten gespeichert. Jede Bootstrap-Aktion protokolliert ihre Ergebnisse in einem separaten, nummerierten Unterverzeichnis: `/mnt/var/log/bootstrap-actions/1/` für die erste Bootstrap-Aktion, `/mnt/var/log/bootstrap-actions/2/` für die zweite, und so weiter.
- **Instance-Statusprotokolle** – Diese Protokolle enthalten Informationen über die CPU, den Arbeitsspeicher und die Garbage Collector-Threads des Knotens. Die Protokolldateien werden in `/mnt/var/log/instance-state/` auf dem Primärknoten gespeichert.

Archivieren von Protokolldateien in Amazon S3

Note

Sie können mit dem `yarn logs`-Dienstprogramm derzeit keine Protokollzusammenführung in Amazon S3 durchführen.

Amazon EMR veröffentlicht 6.9.0 und höher und archiviert Protokolle während der Cluster-Herunterskalierung in Amazon S3, sodass die auf dem Cluster generierten Protokolldateien auch nach dem Beenden des Knotens bestehen bleiben. Dieses Verhalten wird automatisch aktiviert, sodass Sie nichts unternehmen müssen, um es zu aktivieren. Für Amazon-EMR-Versionen 6.8.0 und früher können Sie einen Cluster so konfigurieren, dass die auf dem Primärknoten gespeicherten Protokolldateien regelmäßig in Amazon S3 archiviert werden. Auf diese Weise wird sichergestellt, dass die Protokolldateien verfügbar sind, nachdem der Cluster beendet wird (unabhängig davon, ob dieser normal heruntergefahren wurde oder ob ein Fehler aufgetreten ist). Amazon EMR archiviert die Protokolldateien in 5-Minuten-Intervallen in Amazon S3.

Um die Protokolldateien in Amazon S3 für Amazon-EMR-Versionen 6.8.0 zu archivieren, müssen Sie dieses Feature beim Start des Clusters aktivieren. Sie können dies entweder über Konsole, die CLI oder die API erledigen. Die Protokollierung ist bei über die Konsole gestarteten Clustern standardmäßig aktiviert. Für Cluster, die per CLI oder über die API gestartet wurden, muss die Protokollierung in Amazon S3 manuell aktiviert werden.

Console

Wie Sie Protokolldateien auf Amazon S3 mit der neuen Konsole archivieren

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Aktivieren Sie unter Cluster-Protokolle das Kontrollkästchen Cluster-spezifische Protokolle in Amazon S3 veröffentlichen.
4. Geben Sie im Feld Speicherort von Amazon S3 einen Amazon-S3-Pfad zum Speichern Ihrer Protokolle ein. Wenn Sie den Namen eines Ordners eingeben, der nicht im Bucket vorhanden ist, wird er von Amazon S3 erstellt.

Wenn Sie diesen Wert festlegen, kopiert Amazon EMR die Protokolldateien von den EC2 Instances im Cluster nach Amazon S3. Dadurch wird verhindert, dass die Protokolldateien verloren gehen, wenn der Cluster endet, und beendet dann die EC2 Instances, die den Cluster hosten. Diese Protokolle sind bei der Fehlerbehebung hilfreich. Weitere Informationen finden Sie unter [Protokolldateien anzeigen](#).

5. Aktivieren Sie optional das Kontrollkästchen Clusterspezifische Protokolle verschlüsseln. Wählen Sie dann einen AWS KMS Schlüssel aus der Liste aus, geben Sie einen Schlüssel-

ARN ein oder erstellen Sie einen neuen Schlüssel. Diese Option ist nur mit Amazon-EMR-Version 5.30.0 und höher verfügbar, mit Ausnahme von Version 6.0.0. Um diese Option zu verwenden, fügen Sie Berechtigungen AWS KMS für Ihr EC2 Instance-Profil und Ihre Amazon EMR-Rolle hinzu. Weitere Informationen finden Sie unter [So verschlüsseln Sie Protokolldateien, die Amazon S3 mit einem kundenverwalteten AWS -KMS-Schlüssel gespeichert sind](#).

6. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
7. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um Protokolldateien auf Amazon S3 zu archivieren mit dem AWS CLI

Um Protokolldateien mit dem in Amazon S3 zu archivieren AWS CLI, geben Sie den `create-cluster` Befehl ein und geben Sie den Amazon S3 S3-Protokollpfad mithilfe des `--log-uri` Parameters an.

1. Um Dateien bei Amazon S3 zu protokollieren, geben Sie den folgenden Befehl ein und *myKey* ersetzen Sie ihn durch den Namen Ihres EC2 key pair.

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.9.0 --log-uri s3://DOC-EXAMPLE-BUCKET/logs --applications Name=Hadoop Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

2. Wenn Sie die Instance-Anzahl ohne den `--instance-groups`-Parameter angeben, wird ein einzelner Primärknoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

Note

Wenn Sie noch nicht die standardmäßige Amazon EMR-Servicerolle und das EC2 Instanzprofil erstellt haben, geben Sie ein, um sie `aws emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl eingeben.

So verschlüsseln Sie Protokolldateien, die Amazon S3 mit einem kundenverwalteten AWS -KMS-Schlüssel gespeichert sind

Mit Amazon EMR Version 5.30.0 und höher (außer Amazon EMR 6.0.0) können Sie in Amazon S3 gespeicherte Protokolldateien mit einem vom Kunden verwalteten KMS-Schlüssel verschlüsseln. AWS Um diese Option über die Konsole zu aktivieren, führen Sie die Schritte unter [Archivieren von Protokolldateien in Amazon S3](#) aus. Ihr EC2 Amazon-Instance-Profil und Ihre Amazon EMR-Rolle müssen die folgenden Voraussetzungen erfüllen:

- Das für Ihren Cluster verwendete EC2 Amazon-Instance-Profil muss über eine Nutzungsberechtigung verfügen `kms:GenerateDataKey`.
- Die für den Cluster verwendete Amazon-EMR-Rolle muss über die Berechtigung `kms:DescribeKey` verfügen.
- Das EC2 Amazon-Instance-Profil und die Amazon EMR-Rolle müssen der Liste der Schlüsselbenutzer für den angegebenen kundenverwalteten AWS KMS-Schlüssel hinzugefügt werden, wie die folgenden Schritte zeigen:
 1. Öffnen Sie die Konsole AWS Key Management Service (AWS KMS) unter <https://console.aws.amazon.com/kms>.
 2. Um die AWS Region zu ändern, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
 3. Wählen Sie den Alias des zu ändernden KMS-Schlüssels aus.
 4. Wählen Sie auf der Seite mit den Schlüsseldetails unter Key Users (Schlüsselbenutzer) die Option Add (Hinzufügen) aus.
 5. Wählen Sie im Dialogfeld „Schlüsselbenutzer hinzufügen“ Ihr EC2 Amazon-Instance-Profil und Ihre Amazon EMR-Rolle aus.
 6. Wählen Sie Hinzufügen aus.
- Sie müssen auch den KMS-Schlüssel so konfigurieren, dass die `elasticmapreduce.amazonaws.com` Service Principals `persistentappui.elasticmapreduce.amazonaws.com` und `kms:GenerateDataKey` `kms:GenerateDataKeyWithoutPlaintext` `kms:Decrypt` Auf diese Weise kann EMR mit dem KMS-Schlüssel verschlüsselte Protokolle lesen und in den verwalteten Speicher schreiben. Die Benutzer-IAM-Rolle muss über die Berechtigung zur Verwendung `kms:GenerateDataKey` von und verfügen. `kms:Decrypt`

```
{
```

```

    "Sid": "Allow User Role to use KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "User Role"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "kms:EncryptionContext:aws:elasticmapreduce:clusterId": "j-*",
            "kms:ViaService": "elasticmapreduce.region.amazonaws.com"
        }
    }
},
{
    "Sid": "Allow Persistent APP UI to validate KMS key for write",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "elasticmapreduce.amazonaws.com"
        ]
    },
    "Action": [
        "kms:GenerateDataKeyWithoutPlaintext"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "aws:SourceArn": "arn:aws:elasticmapreduce:region:account:cluster/j-*",
            "kms:EncryptionContext:aws:elasticmapreduce:clusterId": "j-*"
        }
    }
},
{
    "Sid": "Allow Persistent APP UI to Write/Read Logs",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "persistentappui.elasticmapreduce.amazonaws.com",
            "elasticmapreduce.amazonaws.com"
        ]
    }
}

```

```

    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "aws:SourceArn": "arn:aws:elasticmapreduce:region:account:cluster/j-*",
            "kms:EncryptionContext:aws:elasticmapreduce:clusterId": "j-*",
            "kms:ViaService": "s3.region.amazonaws.com"
        }
    }
}

```

Aus Sicherheitsgründen empfehlen wir, die `aws:SourceArn` Bedingungen `kms:EncryptionContext` und hinzuzufügen. Diese Bedingungen tragen dazu bei, dass der Schlüssel nur von Amazon EMR EC2 und nur für Protokolle verwendet wird, die von Aufträgen generiert wurden, die in einem bestimmten Cluster ausgeführt werden.

Weitere Informationen finden Sie unter [Von Amazon EMR verwendete IAM-Service Rollen](#) und [Verwenden von Schlüsselrichtlinien](#) im AWS Key Management Service-Entwicklerhandbuch.

So aggregieren Sie Protokolle in Amazon S3 über die AWS CLI

Note

Sie können mit dem `yarn logs`-Dienstprogramm derzeit keine Protokollzusammenführung durchführen. Sie können die durch dieses Verfahren unterstützte Aggregation nutzen.

Bei der Protokollaggregation (Hadoop 2.x) werden Protokolle für eine bestimmte Anwendung aus allen Containern in einer einzigen Datei zusammengestellt. Um die Protokollaggregation für Amazon S3 mithilfe von zu aktivieren AWS CLI, verwenden Sie beim Clusterstart eine Bootstrap-Aktion, um die Protokollaggregation zu aktivieren und den Bucket zum Speichern der Protokolle anzugeben.

- Um die Protokollaggregation zu aktivieren, erstellen Sie die folgende Konfigurationsdatei mit dem Namen `myConfig.json`, die Folgendes enthält:

```
[
```



```
{
  "Classification": "yarn-site",
  "Properties": {
    "yarn.log-aggregation-enable": "true",
    "yarn.log-aggregation.retain-seconds": "-1",
    "yarn.nodemanager.remote-app-log-dir": "s3://\\DOC-EXAMPLE-BUCKET\\logs"
  }
}
```

Geben Sie den folgenden Befehl ein und *myKey* ersetzen Sie ihn durch den Namen Ihres EC2 key pair. Sie können zusätzlich jeden der roten Texte durch Ihre eigenen Konfigurationen ersetzen.

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.9.0 \
--applications Name=Hadoop \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-type m5.xlarge \
--instance-count 3 \
--configurations file:///myConfig.json
```

Wenn Sie die Instance-Anzahl ohne den `--instance-groups`-Parameter angeben, wird ein einzelner Primärknoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

Note

Wenn Sie noch nicht die standardmäßige EMR-Dienstrolle und das EC2 Instanzprofil erstellt haben, führen Sie zunächst den Befehl aus, um sie `aws emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl ausführen.

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie in der AWS CLI [AWS CLI Befehlsreferenz](#).

Protokollspeicherorte

Die folgende Liste enthält alle Protokolltypen und ihre Speicherorte in Amazon S3. Sie können diese zur Behebung von Problemen mit Amazon EMR verwenden.

Schrittprotokolle

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/steps/<step-id>/
```

Anwendungsprotokolle

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/containers/
```

Dieser Speicherort umfasst Container `stderr` und `stdout`, `directory.info`, `prelaunch.out` und `launch_container.sh`-Protokolle.

Resource-Manager-Protokolle

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/  
applications/hadoop-yarn/
```

Hadoop HDFS

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/  
applications/hadoop-hdfs/
```

Dieser Speicherort umfasst NameNode, DataNode, und TimelineServer YARN-Protokolle.

Knoten-Manager-Protokolle

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/  
applications/hadoop-yarn/
```

Instance-Statusprotokolle

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/daemons/  
instance-state/
```

Bereitstellungsprotokolle für Amazon EMR

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/  
provision-node/*
```

Hive-Protokolle

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/  
applications/hive/*
```

- Um Hive-Protokolle in Ihrem Cluster zu finden, entfernen Sie das Sternchen (*) und fügen Sie `/var/log/hive/` an den obigen Link an.
- Um HiveServer zwei Logs zu finden, entferne das Sternchen (*) und füge es `var/log/hive/hiveserver2.log` an den obigen Link an.
- Um HiveCLI-Protokolle zu finden, entfernen Sie das Sternchen (*) und fügen Sie `/var/log/hive/user/hadoop/hive.log` an den obigen Link an.
- Um Hive-Metastore-Server-Protokolle zu finden, entfernen Sie das Sternchen (*) und fügen Sie `/var/log/hive/user/hive/hive.log` an den obigen Link an.

Wenn Ihr Fehler im Primär- oder Aufgabenknoten Ihrer Tez-Anwendung auftritt, stellen Sie die Protokolle des entsprechenden Hadoop-Containers bereit.

Taggen und kategorisieren Sie Amazon EMR-Cluster-Ressourcen

Es kann praktisch sein, Ihre AWS Ressourcen auf unterschiedliche Weise zu kategorisieren, z. B. nach Zweck, Eigentümer oder Umgebung. Sie können dies in Amazon EMR durch Zuweisen benutzerdefinierter Metadaten zu Ihren Amazon-EMR-Clustern mithilfe von Tags erreichen. Ein Tag besteht aus einem Schlüssel und einem Wert, die Sie beide selbst definieren können. Für Amazon EMR stellt der Cluster die Ressourcenebene dar, die Sie taggen können. Sie können beispielsweise eine Gruppe von Tags für die Cluster Ihres Kontos definieren, mit deren Hilfe Sie den Besitzer der einzelnen Cluster verfolgen oder eine Produktions-Cluster von einem Test-Cluster unterscheiden können. Wir empfehlen das Erstellen eines einheitlichen Satzes von Tags, um Anforderungen Ihres Unternehmens zu erfüllen.

Wenn Sie einem Amazon EMR-Cluster ein Tag hinzufügen, wird das Tag auch an jede aktive EC2 Amazon-Instance weitergegeben, die dem Cluster zugeordnet ist. Wenn Sie ein Tag aus einem Amazon EMR-Cluster entfernen, wird dieses Tag auch von jeder zugehörigen aktiven EC2 Amazon-Instance entfernt.

Important

Verwenden Sie die Amazon EMR-Konsole oder CLI, um Tags auf EC2 Amazon-Instances zu verwalten, die Teil eines Clusters sind, und nicht die EC2 Amazon-Konsole oder CLI, da Änderungen, die Sie in Amazon vornehmen, EC2 nicht wieder mit dem Amazon EMR-Tagging-System synchronisiert werden.

Sie können eine EC2 Amazon-Instance identifizieren, die Teil eines Amazon EMR-Clusters ist, indem Sie nach den folgenden System-Tags suchen. In diesem Beispiel **CORE** ist dies der Wert für die Instance-Gruppenrolle und **j-12345678** ein Beispiel für einen Job-Flow-Identifikationswert (Cluster):

- `aws:elasticmapreduce: = instance-group-role CORE`
- `aws:elasticmapreduce: job-flow-id = j-12345678`

 Note

Amazon EMR und Amazon EC2 interpretieren Ihre Tags als eine Zeichenfolge ohne semantische Bedeutung.

Sie können mithilfe der AWS Management Console, der CLI und der API mit Tags arbeiten.

Sie können Tags beim Erstellen eines neuen Amazon-EMR-Clusters hinzufügen. Außerdem können Sie Tags einem ausgeführten Amazon-EMR-Cluster hinzufügen, in diesem bearbeiten oder aus diesem entfernen. Die Bearbeitung eines Tags bezieht sich auf die Amazon-EMR-Konsole. Wenn Sie die CLI und API zum Bearbeiten eines Tags verwenden, entfernen Sie das alte Tag und fügen ein neues hinzu. Sie können Tag-Schlüssel und Werte bearbeiten und Tags jederzeit aus einer Ressource entfernen, während der Cluster ausgeführt wird. Sie können Tags jedoch nicht hinzufügen, bearbeiten oder aus einem beendeten Cluster oder beendeten Instances entfernen, die zuvor einem Cluster zugeordnet waren, der noch aktiv ist. Darüber hinaus können Sie den Wert eines Tags zwar auf eine leere Zeichenfolge, jedoch nicht auf Null festlegen.

Wenn Sie AWS Identity and Access Management (IAM) mit Ihren EC2 Amazon-Instances für ressourcenbasierte Berechtigungen nach Tag verwenden, werden Ihre IAM-Richtlinien auf Tags angewendet, die Amazon EMR an die Amazon-Instances eines Clusters weitergibt. EC2 Damit Amazon EMR-Tags an Ihre EC2 Amazon-Instances weitergegeben werden können, EC2 muss Ihre IAM-Richtlinie für Amazon Berechtigungen zum Aufrufen von Amazon und zulassen. EC2 CreateTags DeleteTags APIs Außerdem können sich weitergegebene Tags auf die ressourcenbasierten Berechtigungen EC2 Ihres Amazon auswirken. An Amazon weitergegebene Tags EC2 können wie andere EC2 Amazon-Tags als Bedingungen in Ihrer IAM-Richtlinie gelesen werden. Denken Sie an Ihre IAM-Richtlinie, wenn Sie Ihren Amazon-EMR-Clustern Tags hinzufügen, um zu verhindern, dass Benutzer falsche Berechtigungen für einen Cluster erhalten. Stellen Sie sicher, dass Ihre IAM-Richtlinien keine Bedingungen für Tags enthalten, die Sie auch in Ihrem

Amazon-EMR-Cluster verwenden möchten. Andernfalls können Probleme auftreten. Weitere Informationen finden Sie unter [Steuern des Zugriffs auf EC2 Amazon-Ressourcen](#).

Einschränkungen, die für das Markieren von Ressourcen in Amazon EMR gelten

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Einschränkungen, die für EC2 Amazon-Ressourcen gelten, gelten auch für Amazon EMR. Weitere Informationen finden Sie unter https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions.
- Verwenden Sie das `aws:` Präfix nicht in Tag-Namen und -Werten, da es für die AWS Verwendung reserviert ist. Sie können darüber hinaus keine Tag-Namen oder Werte mit diesem Präfix bearbeiten oder löschen.
- Sie können Tags in einem beendeten Cluster nicht ändern oder bearbeiten.
- Ein Tag-Wert kann eine leere Zeichenfolge, aber nicht null sein. Darüber hinaus kann ein Tag-Schlüssel keine leere Zeichenfolge sein.
- Schlüssel und Werte können alphabetische Zeichen in jeder Sprache, numerische Zeichen, Leerzeichen, unsichtbare Trennzeichen und die folgenden Symbole sein: `_ . : / = + - @`.

Weitere Informationen zum Taggen mit dem AWS Management Console finden Sie unter [Arbeiten mit Tags in der Konsole](#) im EC2 Amazon-Benutzerhandbuch. Weitere Informationen zum Tagging mithilfe der EC2 Amazon-API oder der Befehlszeile finden Sie unter [API- und CLI-Übersicht](#) im EC2 Amazon-Benutzerhandbuch.

Amazon EMR-Ressourcen für die Abrechnung kennzeichnen

Sie können Tags verwenden, um Ihre AWS Rechnung so zu organisieren, dass sie Ihre eigene Kostenstruktur widerspiegeln. Melden Sie sich dazu an, um Ihre AWS Kontorechnung mit den Tag-Schlüsselwerten zu erhalten. Anschließend können Sie Ihre Abrechnungsdaten nach Tag-Schlüsselwerten organisieren, um die Kosten kombinierter Ressourcen zu ermitteln. Amazon EMR und Amazon EC2 haben zwar unterschiedliche Abrechnungen, aber die Tags auf jedem Cluster werden auch auf jeder zugehörigen Instance platziert, sodass Sie Tags verwenden können, um zugehörige Amazon EMR- und EC2 Amazon-Kosten miteinander zu verknüpfen.

Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten

dieser Anwendung über mehrere Services hinweg sehen können. Weitere Informationen finden Sie unter [Kostenzuordnung und Tagging](#) im AWS Billing -Benutzerhandbuch.

Hinzufügen von Tags zu einem Amazon EMR-Cluster

Sie können dem Cluster auch Tags hinzufügen, wenn Sie ihn erstellen.

Console

So fügen Sie Tags hinzu, wenn Sie einen Cluster mit der neuen Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Tags die Option Neuen Tag hinzufügen aus. Geben Sie im Feld Schlüssel ein Tag an. Geben Sie optional ein Tag im Feld Wert an.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um Tags hinzuzufügen, wenn Sie einen Cluster mit dem AWS CLI

Im folgenden Beispiel wird gezeigt, wie ein Tag einem neuen Cluster über die AWS CLI hinzugefügt wird. Zum Hinzufügen von Tags beim Erstellen eines Clusters geben Sie den Unterbefehl `create-cluster` mit dem Parameter `--tags` ein.

- Um *marketing* beim Erstellen eines Clusters ein Tag *costCenter* mit einem Schlüsselwert hinzuzufügen, geben Sie den folgenden Befehl ein und *myKey* ersetzen Sie ihn durch den Namen Ihres EC2 key pair.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --  
applications Name=Hadoop Name=Hive Name=Pig --tags "costCenter=marketing" --  
use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --  
instance-count 3
```

Wenn Sie die Instance-Anzahl ohne den Parameter `--instance-groups` angeben, wird ein einzelner Master-Knoten gestartet. Die verbleibenden Instances werden dabei als Core-Knoten gestartet. Alle Knoten verwenden den im Befehl angegebenen Instance-Typ.

 Note

Wenn Sie noch nicht die standardmäßige EMR-Servicerolle und das EC2 Instanzprofil erstellt haben, geben Sie `aws` ein, um sie `emr create-default-roles` zu erstellen, bevor Sie den `create-cluster` Unterbefehl eingeben.

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Sie können Tags auch einem vorhandenen Cluster hinzufügen.

Console

So fügen Sie Tags zu einem vorhandenen Cluster über die neue Konsole hinzu

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Wählen Sie auf der Cluster-Detailseite auf der Registerkarte Tags die Option Tags verwalten aus. Geben Sie im Feld Schlüssel ein Tag an. Geben Sie optional ein Tag im Feld Wert an.
4. Wählen Sie Änderungen speichern aus. Die Registerkarte Tags wird mit der neuen Anzahl von Tags aktualisiert, die Sie in Ihrem Cluster haben. Wenn Sie jetzt beispielsweise zwei Tags haben, lautet die Bezeichnung Ihres Tabs Tags (2).

AWS CLI

Um einem laufenden Cluster Tags hinzuzufügen, verwenden Sie AWS CLI

- Geben Sie den Unterbefehl `add-tags` mit dem Parameter `--tag` ein, um der Cluster-ID Tags zuzuweisen. Sie können die Cluster-ID mithilfe der Konsole oder des Befehls

`list-clusters` finden. Der Unterbefehl `add-tags` akzeptiert derzeit nur einen Ressourcenbezeichner.

Um beispielsweise einem laufenden Cluster zwei Tags hinzuzufügen, eines mit einem Schlüssel mit dem Wert *marketing* und ein anderes *costCenter* mit dem *other* Wert von *accounting*, geben Sie den folgenden Befehl ein und *j-KT4XXXXXXXXX1NM* ersetzen Sie ihn durch Ihre Cluster-ID.

```
aws emr add-tags --resource-id j-KT4XXXXXXXXX1NM --tag "costCenter=marketing" --tag "other=accounting"
```

Beachten Sie, dass beim Hinzufügen von Tags über die AWS CLI keine Ausgabe des Befehls erfolgt. Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Tags auf einem Amazon EMR-Cluster anzeigen

Wenn Sie alle Tags anzeigen möchten, die mit einem Cluster verknüpft sind, können Sie diese in der Konsole oder AWS CLI ansehen.

Console

So zeigen Sie Tags in einem Cluster mit der neuen Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Um alle Ihre Tags anzuzeigen, wählen Sie auf der Cluster-Detailseite die Registerkarte Tags aus.

AWS CLI

Um Tags auf einem Cluster anzuzeigen, verwenden Sie AWS CLI

Um die Tags auf einem Cluster mithilfe von anzuzeigen AWS CLI, geben Sie den `describe-cluster` Unterbefehl mit dem `--query` Parameter ein.

- Um die Tags eines Clusters anzuzeigen, geben Sie den folgenden Befehl ein und **j-KT4XXXXXXXX1NM** ersetzen Sie ihn durch Ihre Cluster-ID.

```
aws emr describe-cluster --cluster-id j-KT4XXXXXXXX1NM --query Cluster.Tags
```

Die Ausgabe enthält alle Tag-Informationen über den Cluster ähnlich wie diese:

```
Value: accounting      Value: marketing
Key: other             Key: costCenter
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Tags aus einem Amazon EMR-Cluster entfernen

Wenn Sie ein Tag nicht mehr benötigen, können Sie es aus dem Cluster entfernen.

Console

So entfernen Sie Tags in einem Cluster mit der neuen Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Wählen Sie auf der Cluster-Detailseite auf der Registerkarte Tags die Option Tags verwalten aus.
4. Wählen Sie Entfernen für jedes Schlüssel-Wert-Paar, das Sie entfernen möchten.
5. Wählen Sie Änderungen speichern aus.

AWS CLI

Um Tags auf einem Cluster zu entfernen, verwenden Sie AWS CLI

Geben Sie den `remove-tags`-Unterbefehl mit dem `--tag-keys`-Parameter ein. Beim Entfernen eines Tags ist nur der Schlüsselname erforderlich.

- Um ein Tag aus einem Cluster zu entfernen, geben Sie den folgenden Befehl ein und **j-KT4XXXXXXXX1NM** ersetzen Sie ihn durch Ihre Cluster-ID.

```
aws emr remove-tags --resource-id j-KT4XXXXXXXX1NM --tag-keys "costCenter"
```

 Note

Sie können derzeit nicht mehrere Tags mit einem einzigen Befehl entfernen.

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Integration von Treibern und Drittanbieteranwendungen auf Amazon EMR

Sie können verschiedene beliebte Big-Data-Anwendungen in Amazon EMR zu Preisen eines Dienstprogramms ausführen. Das bedeutet, dass Sie eine geringe zusätzliche Gebühr pro Stunde für die Drittanbieter-Anwendung zahlen, während der Cluster ausgeführt wird. So können Sie die Anwendung nutzen, ohne eine Jahreslizenz erwerben zu müssen. Die folgenden Abschnitte beschreiben einige der Tools, die Sie mit EMR verwenden können.

Themen

- [Verwenden von Business-Intelligence-Tools in Amazon EMR](#)

Verwenden von Business-Intelligence-Tools in Amazon EMR

Sie können beliebte Business Intelligence-Tools wie Microsoft Excel, MicroStrategyQlikView, und Tableau mit Amazon EMR verwenden, um Ihre Daten zu untersuchen und zu visualisieren. Viele dieser Tools erfordern einen ODBC- (Open Database Connectivity) oder JDBC-Treiber (Java Database Connectivity). Informationen zum Herunterladen und Installieren der neuesten Treiber finden Sie unter <http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/>.

Ältere Versionen von Treibern finden Sie unter <http://awssupportdatasvcs.com/bootstrap-actions/Simba/>.

Sicherheit in Amazon EMR

Sicherheit und Compliance sind eine Verantwortung, mit der Sie sich teilen. AWS Dieses Modell der geteilten Verantwortung kann Ihnen helfen, Ihre betriebliche Belastung zu verringern, da AWS die Komponenten vom Host-Betriebssystem und der Virtualisierungsebene bis hin zur physischen Sicherheit der Einrichtungen, in denen EMR-Cluster betrieben werden, betrieben, verwaltet und kontrolliert werden. Sie übernehmen die Verantwortung für die Verwaltung und Aktualisierung von Amazon EMR-Clustern sowie die Konfiguration der Anwendungssoftware und die AWS bereitgestellten Sicherheitskontrollen. Diese Differenzierung der Verantwortung wird allgemein als Sicherheit in der Cloud und Sicherheit in der Cloud bezeichnet.

- **Sicherheit der Cloud** — AWS ist verantwortlich für den Schutz der Infrastruktur, AWS-Services in der sie ausgeführt wird AWS. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS -Compliance-Programme](#) regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon EMR gelten, finden Sie [AWS-Services unter Umfang nach Compliance-Programmen](#).
- **Sicherheit in der Cloud** — Sie sind auch dafür verantwortlich, alle erforderlichen Sicherheitskonfigurations- und Verwaltungsaufgaben zur Sicherung eines Amazon EMR-Clusters durchzuführen. Kunden, die einen Amazon EMR-Cluster bereitstellen, sind für die Verwaltung der auf den Instances installierten Anwendungssoftware und die Konfiguration der AWS bereitgestellten Funktionen wie Sicherheitsgruppen, Verschlüsselung und Zugriffskontrolle gemäß Ihren Anforderungen, geltenden Gesetzen und Vorschriften verantwortlich.

Diese Dokumentation zeigt Ihnen, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon EMR einsetzen können. Die Themen in diesem Kapitel zeigen Ihnen, wie Sie Amazon EMR konfigurieren und andere verwenden, AWS-Services um Ihre Sicherheits- und Compliance-Ziele zu erreichen.

Netzwerk- und Infrastruktursicherheit

Als verwalteter Service ist Amazon EMR durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper [Amazon Web Services: Sicherheitsprozesse im Überblick](#) beschrieben sind. AWS Die Dienste zum Schutz von Netzwerken und Infrastrukturen bieten Ihnen differenzierten Schutz sowohl auf Host- als auch auf Netzwerkebene. Amazon EMR unterstützt AWS-Services

und bietet Anwendungsfunktionen, die Ihren Netzwerkschutz- und Compliance-Anforderungen entsprechen.

- EC2 Amazon-Sicherheitsgruppen fungieren als virtuelle Firewall für Amazon EMR-Cluster-Instances und begrenzen den eingehenden und ausgehenden Netzwerkverkehr. Weitere Informationen finden Sie unter [Steuern des Netzwerkverkehrs mit Sicherheitsgruppen](#).
- Amazon EMR Block Public Access (BPA) verhindert, dass Sie einen Cluster in einem öffentlichen Subnetz starten, wenn der Cluster über eine Sicherheitskonfiguration verfügt, die eingehenden Datenverkehr von öffentlichen IP-Adressen an einem Port zulässt. Weitere Informationen finden Sie unter [Verwenden von Amazon EMR, um den öffentlichen Zugriff zu blockieren](#).
- Secure Shell (SSH) bietet Benutzern eine sichere Möglichkeit, eine Verbindung zur Befehlszeile auf Cluster-Instances herzustellen. Sie können SSH auch verwenden, um Webschnittstellen anzuzeigen, die Anwendungen auf dem Master-Knoten eines Clusters hosten. Weitere Informationen finden Sie unter [Verwenden eines EC2 key pair für SSH-Anmeldeinformationen](#) und [Connect zu einem Cluster](#) herstellen.

Updates des standardmäßigen Amazon Linux AMI für Amazon EMR

Important

EMR-Cluster, auf denen Amazon Linux oder Amazon Linux 2 Amazon Machine Images (AMIs) ausgeführt werden, verwenden das Standardverhalten von Amazon Linux und laden wichtige und kritische Kernel-Updates, die einen Neustart erfordern, nicht automatisch herunter und installieren sie. Dies ist dasselbe Verhalten wie bei anderen EC2 Amazon-Instances, auf denen das standardmäßige Amazon Linux-AMI ausgeführt wird. Wenn neue Amazon-Linux-Softwareupdates, die einen Neustart erfordern (wie Kernel-, NVIDIA- und CUDA-Updates), nach der Veröffentlichung einer Amazon-EMR-Version verfügbar werden, laden EMR-Cluster-Instances, die das Standard-AMI ausführen, diese Updates nicht automatisch herunter und installieren sie. Um Kernel-Updates zu erhalten, können Sie [Ihr Amazon-EMR-AMI](#) so anpassen, dass es [das neueste Amazon-Linux-AMI verwendet](#).

Abhängig von der Sicherheit Ihrer Anwendung und der Dauer der Ausführung eines Clusters können Sie wählen, ob Sie Ihr Cluster regelmäßig neu starten, um Sicherheitsupdates anzuwenden, oder ob Sie eine Bootstrap-Aktion zum Anpassen von Paketinstallation und Updates erstellen. Sie können

außerdem Sicherheitsupdates erst testen und dann auf ausgeführten Cluster-Instances installieren. Weitere Informationen finden Sie unter [Verwenden des Standard-Amazon-Linux-AMI für Amazon EMR](#). Beachten Sie, dass Ihre Netzwerkkonfiguration den HTTP- und HTTPS-Ausgang zu Linux-Repositorys in Amazon S3 zulassen muss, da andernfalls Sicherheitsupdates nicht erfolgreich sein werden.

AWS Identity and Access Management mit Amazon EMR

AWS Identity and Access Management (IAM) ist ein AWS Service, der einem Administrator hilft, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon-EMR-Ressourcen zu nutzen. IAM-Identitäten umfassen Benutzer, Gruppen und Rollen. Eine IAM-Rolle ähnelt einem IAM-Benutzer, ist jedoch keiner bestimmten Person zugeordnet und soll von jedem Benutzer übernommen werden können, der Berechtigungen benötigt. Weitere Informationen finden Sie unter [AWS Identity and Access Management Amazon EMR](#). Amazon EMR verwendet mehrere IAM-Rollen, um Sie bei der Implementierung von Zugriffskontrollen für Amazon EMR-Cluster zu unterstützen. IAM ist ein AWS Service, den Sie ohne zusätzliche Kosten nutzen können.

- IAM-Rolle für Amazon EMR (EMR-Rolle) — steuert, wie der Amazon EMR-Service in Ihrem Namen AWS-Services auf andere zugreifen kann, z. B. die Bereitstellung von EC2 Amazon-Instances beim Start des Amazon EMR-Clusters. Weitere Informationen finden [Sie unter Konfigurieren von IAM-Servicerollen für Amazon EMR-Berechtigungen AWS-Services und Ressourcen](#).
- IAM-Rolle für EC2 Cluster-Instances (EC2 Instance-Profil) — eine Rolle, die jeder EC2 Instance im Amazon EMR-Cluster zugewiesen wird, wenn die Instance gestartet wird. Anwendungsprozesse, die auf dem Cluster ausgeführt werden, verwenden diese Rolle, um mit anderen zu interagieren AWS-Services, z. B. mit Amazon S3. Weitere Informationen finden Sie unter [IAM-Rolle für EC2 Cluster-Instances](#).
- IAM-Rolle für Anwendungen (Runtime-Rolle) — eine IAM-Rolle, die Sie angeben können, wenn Sie einen Job oder eine Anfrage an einen Amazon EMR-Cluster senden. Der Job oder die Abfrage, die Sie an Ihren Amazon EMR-Cluster senden, verwendet die Runtime-Rolle, um auf AWS Ressourcen wie Objekte in Amazon S3 zuzugreifen. Sie können Laufzeit-Rollen mit Amazon EMR für Spark- und Hive-Jobs angeben. Mithilfe von Runtime-Rollen können Sie Jobs, die auf demselben Cluster ausgeführt werden, mithilfe verschiedener IAM-Rollen isolieren. Weitere Informationen finden Sie unter [Verwenden der IAM-Rolle als Runtime-Rolle mit Amazon EMR](#).

Personalidentitäten beziehen sich auf Benutzer, in denen Workloads erstellt oder ausgeführt werden. AWS Amazon EMR bietet Unterstützung für Mitarbeiteridentitäten mit folgenden Funktionen:

- AWS Das IAM Identity Center (Idc) wird AWS-Service für die Verwaltung des Benutzerzugriffs auf Ressourcen empfohlen. AWS Es ist ein zentraler Ort, an dem Sie die Identitäten Ihrer Belegschaft zuweisen können, sodass Sie konsistent auf mehrere AWS Konten und Anwendungen zugreifen können. Amazon EMR unterstützt die Identitäten von Mitarbeitern durch vertrauenswürdige Weitergabe von Identitäten. Mit der Funktion zur Weitergabe vertrauenswürdiger Identitäten kann sich ein Benutzer bei der Anwendung anmelden, und diese Anwendung kann die Identität des Benutzers an andere AWS-Services weitergeben, um den Zugriff auf Daten oder Ressourcen zu autorisieren. Weitere Informationen finden Sie unter Unterstützung für [AWS IAM Identity Center mit Amazon EMR](#) aktivieren.

Das Lightweight Directory Access Protocol (LDAP) ist ein offenes, herstellerneutrales, branchenübliches Anwendungsprotokoll für den Zugriff auf und die Verwaltung von Informationen über Benutzer, Systeme, Dienste und Anwendungen über das Netzwerk. LDAP wird häufig für die Benutzerauthentifizierung gegen Unternehmensidentitätsserver wie Active Directory (AD) und OpenLDAP verwendet. Durch die Aktivierung von LDAP mit EMR-Clustern ermöglichen Sie es Ihren Benutzern, ihre vorhandenen Anmeldeinformationen für die Authentifizierung und den Zugriff auf Cluster zu verwenden. Weitere Informationen finden Sie unter [Aktivieren der Unterstützung für LDAP mit Amazon EMR](#).

Kerberos ist ein Netzwerkauthentifizierungsprotokoll, das entwickelt wurde, um mithilfe von Secret-Key-Kryptografie eine starke Authentifizierung für client/server Anwendungen zu ermöglichen. Wenn Sie Kerberos verwenden, konfiguriert Amazon EMR Kerberos für die Anwendungen, Komponenten und Subsysteme, die es auf dem Cluster installiert, sodass sie sich gegenseitig authentifizieren. Für den Zugriff auf einen Cluster mit konfiguriertem Kerberos muss ein Kerberos-Prinzipal im Kerberos Domain Controller (KDC) vorhanden sein. Weitere Informationen finden Sie unter [Unterstützung für Kerberos mit Amazon EMR aktivieren](#).

Cluster mit einem Mandanten und mehreren Mandanten

Ein Cluster ist standardmäßig für einen einzelnen Mandanten mit dem EC2 Instanzprofil als IAM-Identität konfiguriert. In einem Single-Tenant-Cluster hat jeder Job vollen und vollständigen Zugriff auf den Cluster, und der Zugriff auf alle AWS-Services Ressourcen erfolgt auf der Grundlage des EC2 Instanzprofils. In einem Multi-Tenant-Cluster sind die Mandanten voneinander isoliert und die Mandanten haben keinen vollen und vollständigen Zugriff auf die Cluster und EC2 Instanzen

des Clusters. Bei Clustern mit mehreren Mandanten werden entweder die Runtime-Rollen oder die Belegschaft identifiziert. In einem Multi-Tenant-Cluster können Sie auch die Unterstützung für feinkörnige Zugriffskontrolle (FGAC) über oder Apache Ranger aktivieren. AWS Lake Formation Bei einem Cluster, für den Runtime-Rollen oder FGAC aktiviert sind, ist der Zugriff auf das EC2 Instanzprofil ebenfalls über iptables deaktiviert.

Important

Jeder Benutzer, der Zugriff auf einen Single-Tenant-Cluster hat, kann jede Software auf dem Linux-Betriebssystem (OS) installieren, von Amazon EMR installierte Softwarekomponenten ändern oder entfernen und sich auf die EC2 Instances auswirken, die Teil des Clusters sind. Wenn Sie sicherstellen möchten, dass Benutzer keine Konfigurationen eines Amazon EMR-Clusters installieren oder ändern können, empfehlen wir Ihnen, die Mehrmandantenfähigkeit für den Cluster zu aktivieren. Sie können Multi-Tenancy auf einem Cluster aktivieren, indem Sie die Unterstützung für Runtime Role, AWS IAM Identity Center, Kerberos oder LDAP aktivieren.

Datenschutz

Mit können Sie Ihre Daten kontrollieren AWS, indem Sie Tools verwenden AWS-Services , um zu bestimmen, wie die Daten gesichert sind und wer Zugriff darauf hat. Mit Diensten wie AWS Identity and Access Management (IAM) können Sie den Zugriff auf AWS-Services und Ressourcen sicher verwalten. AWS CloudTrail ermöglicht Erkennung und Prüfung. Amazon EMR macht es Ihnen leicht, ruhende Daten in Amazon S3 zu verschlüsseln, indem Sie Schlüssel verwenden, die entweder von Ihnen verwaltet werden AWS oder vollständig von Ihnen verwaltet werden. Amazon EMR unterstützt auch die Aktivierung der Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsseln von Daten im Ruhezustand und bei der Übertragung](#).

Datenzugriffskontrolle

Mit der Datenzugriffskontrolle können Sie steuern, auf welche Daten eine IAM-Identität oder eine Mitarbeiteridentität zugreifen kann. Amazon EMR unterstützt die folgenden Zugriffskontrollen:

- Identitätsbasierte IAM-Richtlinien — verwalten Sie Berechtigungen für IAM-Rollen, die Sie mit Amazon EMR verwenden. IAM-Richtlinien können mit Tagging kombiniert werden, um den Zugriff auf einer bestimmten Grundlage zu kontrollieren. cluster-by-cluster Weitere Informationen finden Sie unter [AWS Identity and Access Management Amazon EMR](#).

- AWS Lake Formation zentralisiert die Rechteverwaltung Ihrer Daten und erleichtert die gemeinsame Nutzung innerhalb Ihrer Organisation und extern. Sie können Lake Formation verwenden, um einen detaillierten Zugriff auf Spaltenebene auf Datenbanken und Tabellen im Glue-Datenkatalog zu ermöglichen. AWS Weitere Informationen finden Sie unter [Verwenden AWS Lake Formation mit Amazon EMR](#).
- Amazon S3 Access ermöglicht Zuordnungsidentitäten, Identitäten in Verzeichnissen wie Active Directory oder AWS Identity and Access Management (IAM) -Prinzipalen Datensätzen in S3 zuzuordnen. Darüber hinaus gewährt der S3-Zugriff die Protokollierung der Endbenutzeridentität und der Anwendung, die für den Zugriff auf S3-Daten verwendet wird. AWS CloudTrail Weitere Informationen finden Sie unter [Verwenden von Amazon S3 S3-Zugriffsberechtigungen mit Amazon EMR](#).
- Apache Ranger ist ein Framework zur Aktivierung, Überwachung und Verwaltung einer umfassenden Datensicherheit auf der gesamten Hadoop-Plattform. Amazon EMR unterstützt eine auf Apache Ranger basierende, feinkörnige Zugriffskontrolle für Apache Hive Metastore und Amazon S3. Weitere Informationen finden Sie unter [Integrieren von Apache Ranger in Amazon EMR](#).

Verwenden Sie Sicherheitskonfigurationen, um die Amazon EMR-Cluster-Sicherheit einzurichten

Sie können Amazon-EMR-Sicherheitskonfigurationen verwenden, um Datenverschlüsselung, Kerberos-Authentifizierung und Amazon-S3-Autorisierung für EMRFS auf Ihren Clustern zu konfigurieren. Zunächst erstellen Sie eine Sicherheitskonfiguration. Anschließend kann die Sicherheitskonfiguration bei der Erstellung von wiederholt verwendet werden.

Sie können die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die verwenden, um Sicherheitskonfigurationen AWS SDKs zu erstellen. Sie können auch eine AWS CloudFormation Vorlage verwenden, um eine Sicherheitskonfiguration zu erstellen. Weitere Informationen finden Sie im [AWS CloudFormation Benutzerhandbuch](#) und in der Vorlagenreferenz für [AWS::EMR::SecurityConfiguration](#).

Themen

- [Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI](#)
- [Geben Sie eine Sicherheitskonfiguration für einen Amazon EMR-Cluster an](#)

Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI

Dieses Thema behandelt allgemeine Verfahren zum Erstellen einer Sicherheitskonfiguration mit der Amazon EMR-Konsole und der AWS CLI, gefolgt von einer Referenz zu den Parametern, die Verschlüsselung, Authentifizierung und IAM-Rollen für EMRFS umfassen. Weitere Informationen zu diesen Funktionen finden Sie in den folgenden Themen:

- [Verschlüsseln Sie Daten im Ruhezustand und bei der Übertragung mit Amazon EMR](#)
- [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#)
- [Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3](#)

So erstellen Sie eine Sicherheitskonfiguration mithilfe der Konsole

1. Öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie im Navigationsbereich Security Configurations (Sicherheitskonfigurationen), Create security configuration (Sicherheitskonfiguration erstellen) aus.
3. Geben Sie in Name (Name) einen Namen für die Sicherheitskonfiguration ein.
4. Wählen Sie Optionen für Verschlüsselung, und Authentifizierung aus wie in den folgenden Abschnitten beschrieben. Wählen Sie anschließend Erstellen aus.

Um eine Sicherheitskonfiguration mit dem zu erstellen AWS CLI

- Verwenden Sie den Befehl `create-security-configuration` wie im folgenden Beispiel gezeigt.
 - Geben Sie für *SecConfigName* den Namen der Sicherheitskonfiguration an. Dies ist der Name, den Sie angeben, wenn Sie einen Cluster erstellen, der diese Sicherheitskonfiguration verwendet.
 - Geben Sie in *SecConfigDef* eine Inline-JSON-Struktur oder den Pfad zu einer lokalen JSON-Datei an, z. B. *file://MySecConfig.json*. Die JSON-Parameter definieren Optionen für Verschlüsselung, IAM Rollen für EMRFS-Zugriff auf Amazon S3 und Authentifizierung, wie in den folgenden Abschnitten beschrieben.

```
aws emr create-security-configuration --name "SecConfigName" --security-configuration SecConfigDef
```

Datenverschlüsselung konfigurieren

Bevor Sie die Verschlüsselung in einer Sicherheitskonfiguration konfigurieren, erstellen Sie die Schlüssel und Zertifikate, die für die Verschlüsselung verwendet werden. Weitere Informationen erhalten Sie unter [Bereitstellung von Schlüsseln für die Verschlüsselung von Daten im Ruhezustand](#) und [Bereitstellen von Zertifikaten für die Verschlüsselung von Daten während der Übertragung mit der Amazon-EMR-Verschlüsselung](#).

Beim Erstellen einer Sicherheits-Konfiguration legen Sie zwei Verschlüsselungsoptionen fest: Verschlüsselung von Daten während der Übertragung und im Ruhezustand. Die Optionen für die Datenverschlüsselung im Ruhezustand umfassen sowohl die Amazon S3-Verschlüsselung mit EMRFS und die lokale Laufwerksverschlüsselung. Die Optionen für die Verschlüsselung von Daten während der Übertragung aktivieren die Open-Source-Verschlüsselungsfunktionen für bestimmte Anwendungen, die Transport Layer Security (TLS) unterstützen. Die Optionen für die Verschlüsselung während der Übertragung und im Ruhezustand können gemeinsam oder einzeln aktiviert werden. Weitere Informationen finden Sie unter [Verschlüsseln Sie Daten im Ruhezustand und bei der Übertragung mit Amazon EMR](#).

Note

Bei der Nutzung AWS KMS fallen Gebühren für die Speicherung und Verwendung von Verschlüsselungsschlüsseln an. Weitere Informationen finden Sie unter [AWS KMS – Preise](#).

Angeben von Verschlüsselungsoptionen mit der Konsole

Wählen Sie Optionen unter Encryption (Verschlüsselung) entsprechend den folgenden Anleitungen aus.

- Wählen Sie Optionen unter At rest encryption (Verschlüsselung im Ruhezustand) aus, um innerhalb des Dateisystems gespeicherte Daten zu verschlüsseln.

Sie können Daten in Amazon S3, auf lokalen Datenträgern oder in beiden Speichern verschlüsseln.

- Unter S3-Datenverschlüsselung, für die Option Verschlüsselungsmodus wählen Sie einen Wert aus, der festlegt, wie Amazon EMR; die Amazon-S3-Daten mit EMRFS verschlüsselt.

Der nächste Schritt hängt von dem von Ihnen gewählten Verschlüsselungsmodus ab:

- SSE-S3 (SSE-S3)

Angaben zur [serverseitigen Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln](#). Sie müssen nicht mehr tun, da Amazon S3 die Handhabung der Schlüssel für Sie übernimmt.

- SSE-KMS (SSE-KMS) oder CSE-KMS (CSE-KMS)

Gibt [serverseitige Verschlüsselung mit AWS KMS verwalteten Schlüsseln \(SSE-KMS\)](#) oder [clientseitige Verschlüsselung mit AWS KMS](#) verwalteten Schlüsseln (CSE-KMS) an. Wählen Sie für AWS KMS key einen Schlüssel aus. Der Schlüssel muss sich in derselben Region befinden wie Ihr EMR-Cluster. Schlüsselanforderungen finden Sie unter [AWS KMS keys Für die Verschlüsselung verwenden](#).

- CSE-Custom (CSE-Custom)

Gibt [clientseitige Verschlüsselung mit einem benutzerdefinierten clientseitigen Masterschlüssel \(CSE-Custom\) an](#). Geben Sie für das S3-Objekt den Speicherort Ihrer benutzerdefinierten Schlüsselanbieter-JAR-Datei in Amazon S3 oder den Amazon-S3-ARN ein. Geben Sie dann für Key Provider Class den vollständigen Klassennamen einer Klasse ein, die in Ihrer Anwendung deklariert ist, die die Schnittstelle implementiert. EncryptionMaterialsProvider

- Wählen Sie unter Local disk encryption (Lokale Laufwerksverschlüsselung) einen Wert für Key provider type (Schlüsselanbieterart) aus.
 - AWS KMS key

Wählen Sie diese Option, um eine AWS KMS key anzugeben. Wählen Sie für AWS KMS key einen Schlüssel aus. Der Schlüssel muss sich in derselben Region befinden wie Ihr EMR-Cluster. Weitere Informationen zu den Anforderungen für Schlüssel finden Sie unter [AWS KMS keys Für die Verschlüsselung verwenden](#).

EBS Encryption (EBS-Verschlüsselung)

Wenn Sie dies AWS KMS als Ihren Schlüsselanbieter angeben, können Sie die EBS-Verschlüsselung aktivieren, um das EBS-Stammgerät und die Speichervolumes zu verschlüsseln. Um diese Option zu aktivieren, müssen Sie der Amazon-EMR-Servicerolle `EMR_DefaultRole` Berechtigungen zur Verwendung des von Ihnen angegebenen AWS KMS key erteilen. Weitere Informationen zu den Anforderungen für Schlüssel finden Sie unter

Aktivieren der EBS-Verschlüsselung durch Bereitstellung zusätzlicher Berechtigungen für KMS-Schlüssel.

- Custom (Benutzerdefiniert)

Wählen Sie diese Option aus, um einen benutzerdefinierten Schlüsselanbieter festzulegen. Geben Sie für das S3-Objekt den Speicherort Ihrer benutzerdefinierten Schlüsselanbieter-JAR-Datei in Amazon S3 oder den Amazon-S3-ARN ein. Geben Sie für Key Provider Class den vollständigen Klassennamen einer Klasse ein, die in Ihrer Anwendung deklariert ist, die die Schnittstelle implementiert. EncryptionMaterialsProvider Der Klassename, den Sie hier angeben, muss sich von dem Klassennamen für CSE-Custom unterscheiden.

- Wählen Sie In-transit encryption (Verschlüsselung bei Übertragung) aus, um die Open-Source-TLS-Verschlüsselungsfunktionen für Daten während der Übertragung zu aktivieren. Wählen Sie anhand der folgenden Anleitungen einen Certificate provider type (Zertifikatanbietertyp) aus:

- PEM (PEM)

Wählen Sie diese Option zur Verwendung von PEM-Dateien aus, die Sie in einer ZIP-Datei bereitstellen. Zwei Artefakte sind innerhalb der ZIP-Datei erforderlich: privateKey.pem und certificateChain.pem. Eine dritte Datei, trustedCertificates.pem, ist optional. Details dazu finden Sie unter [Bereitstellen von Zertifikaten für die Verschlüsselung von Daten während der Übertragung mit der Amazon-EMR-Verschlüsselung](#). Geben Sie in S3-Objekt den Speicherort in Amazon S3 oder den Amazon-S3-ARN des ZIP-Datei-Felds an.

- Custom (Benutzerdefiniert)

Wählen Sie diese Option aus, um einen benutzerdefinierten Zertifikatanbieter anzugeben. Geben Sie anschließend in S3-Objekt den Speicherort in Amazon S3 oder den Amazon-S3-ARN der JAR-Datei Ihres benutzerdefinierten Zertifikatanbieters ein. Geben Sie für Key Provider Class den vollständigen Klassennamen einer Klasse ein, die in Ihrer Anwendung deklariert ist und die TLSArtifacts Provider-Schnittstelle implementiert.

Angeben von Verschlüsselungsoptionen mit dem AWS CLI

Die folgenden Abschnitte enthalten Beispielszenarien mit ordnungsgemäß formatiertem --security-configuration JSON für verschiedene Konfigurationen und Schlüsselanbieter, gefolgt von einer Referenz für JSON-Parameter und geeignete Werte.

Beispiel der Datenverschlüsselungsoptionen während der Übertragung

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist aktiviert und die Verschlüsselung von Daten im Ruhezustand ist deaktiviert.
- Eine ZIP-Datei mit Zertifikaten in Amazon S3 wird als Schlüsselanbieter verwendet (die Zertifikatanforderungen finden Sie unter [Bereitstellen von Zertifikaten für die Verschlüsselung von Daten während der Übertragung mit der Amazon-EMR-Verschlüsselung](#)).

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": false,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "PEM",
        "S3Object": "s3://MyConfigStore/artifacts/MyCerts.zip"
      }
    }
  }
}'
```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist aktiviert und die Verschlüsselung von Daten im Ruhezustand ist deaktiviert.
- Ein benutzerdefinierter Schlüsselanbieter wird verwendet (die Zertifikatanforderungen finden Sie unter [Bereitstellen von Zertifikaten für die Verschlüsselung von Daten während der Übertragung mit der Amazon-EMR-Verschlüsselung](#)).

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": false,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "Custom",
        "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
        "CertificateProviderClass": "com.mycompany.MyCertProvider"
      }
    }
  }
}'
```

```

    }
  }
}
}'

```

Beispiel der Datenverschlüsselungsoptionen im Ruhezustand

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist deaktiviert und die Verschlüsselung von Daten im Ruhezustand ist aktiviert.
- Für die Amazon-S3-Verschlüsselung wird SSE-S3 verwendet.
- Die lokale Festplattenverschlüsselung wird AWS KMS als Schlüsselanbieter verwendet.

```

aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": false,
    "EnableAtRestEncryption": true,
    "AtRestEncryptionConfiguration": {
      "S3EncryptionConfiguration": {
        "EncryptionMode": "SSE-S3"
      },
      "LocalDiskEncryptionConfiguration": {
        "EncryptionKeyProviderType": "AwsKms",
        "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      }
    }
  }
}'

```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Datenverschlüsselung während der Übertragung ist aktiviert und verweist unter Verwendung der ARN auf eine ZIP-Datei mit PEM-Zertifikaten in Amazon S3.
- Für die KMS-Verschlüsselung wird SSE-Amazon S3 verwendet.
- Die lokale Festplattenverschlüsselung wird AWS KMS als Schlüsselanbieter verwendet.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": true,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "PEM",
        "S3Object": "arn:aws:s3:::MyConfigStore/artifacts/MyCerts.zip"
      }
    },
    "AtRestEncryptionConfiguration": {
      "S3EncryptionConfiguration": {
        "EncryptionMode": "SSE-KMS",
        "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      },
      "LocalDiskEncryptionConfiguration": {
        "EncryptionKeyProviderType": "AwsKms",
        "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      }
    }
  }
}'
```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Datenverschlüsselung während der Übertragung ist aktiviert und verweist auf eine ZIP-Datei mit PEM-Zertifikaten in Amazon S3.
- Für die Amazon-S3-Verschlüsselung wird CSE-KMS verwendet.
- Die lokale Laufwerksverschlüsselung verwendet einen benutzerdefinierten Schlüsselanbieter, auf den anhand des ARN verwiesen wird.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": true,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "PEM",
```

```

    "S3object": "s3://MyConfigStore/artifacts/MyCerts.zip"
  },
  "AtRestEncryptionConfiguration": {
    "S3EncryptionConfiguration": {
      "EncryptionMode": "CSE-KMS",
      "AwsKmsKey": "arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
    },
    "LocalDiskEncryptionConfiguration": {
      "EncryptionKeyProviderType": "Custom",
      "S3object": "arn:aws:s3:::artifacts/MyKeyProvider.jar",
      "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
    }
  }
}
}'

```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung anhand eines benutzerdefinierten Schlüsselanbieters ist aktiviert.
- CSE-Custom wird für Amazon-S3-Daten verwendet.
- Die lokale Laufwerksverschlüsselung verwendet einen benutzerdefinierten Schlüsselanbieter.

```

aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": "true",
    "EnableAtRestEncryption": "true",
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "Custom",
        "S3object": "s3://MyConfig/artifacts/MyCerts.jar",
        "CertificateProviderClass": "com.mycompany.MyCertProvider"
      }
    }
  },
  "AtRestEncryptionConfiguration": {
    "S3EncryptionConfiguration": {
      "EncryptionMode": "CSE-Custom",
      "S3object": "s3://MyConfig/artifacts/MyCerts.jar",

```



```

    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
  },
  "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
  }
}
}
}'

```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist deaktiviert und die Verschlüsselung von Daten im Ruhezustand ist aktiviert.
- Die Amazon S3-Verschlüsselung ist mit SSE-KMS aktiviert.
- Es werden mehrere AWS KMS Schlüssel verwendet, einer für jeden S3-Bucket, und Verschlüsselungsausnahmen werden auf diese einzelnen S3-Buckets angewendet.
- Die lokale Laufwerksverschlüsselung ist deaktiviert.

```

aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "AtRestEncryptionConfiguration": {
      "S3EncryptionConfiguration": {
        "EncryptionMode": "SSE-KMS",
        "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012",
        "Overrides": [
          {
            "BucketName": "amzn-s3-demo-bucket1",
            "EncryptionMode": "SSE-S3"
          },
          {
            "BucketName": "amzn-s3-demo-bucket2",
            "EncryptionMode": "CSE-KMS",
            "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
          },
          {
            "BucketName": "amzn-s3-demo-bucket3",
            "EncryptionMode": "SSE-KMS",

```

```

        "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
    }
]
},
"EnableInTransitEncryption": false,
"EnableAtRestEncryption": true
}
}'

```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist deaktiviert und die Verschlüsselung von Daten im Ruhezustand ist aktiviert.
- Die Amazon S3-Verschlüsselung wird mit SSE-S3 aktiviert und die lokale Laufwerksverschlüsselung ist deaktiviert.

```

aws emr create-security-configuration --name "MyS3EncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
        "AtRestEncryptionConfiguration": {
            "S3EncryptionConfiguration": {
                "EncryptionMode": "SSE-S3"
            }
        }
    }
}'

```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist deaktiviert und die Verschlüsselung von Daten im Ruhezustand ist aktiviert.
- Die lokale Festplattenverschlüsselung ist AWS KMS als Schlüsselanbieter aktiviert und die Amazon S3 S3-Verschlüsselung ist deaktiviert.

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": false,
    "EnableAtRestEncryption": true,
    "AtRestEncryptionConfiguration": {
      "LocalDiskEncryptionConfiguration": {
        "EncryptionKeyProviderType": "AwsKms",
        "AwsKmsKey": "arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      }
    }
  }
}'
```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- Die Verschlüsselung von Daten während der Übertragung ist deaktiviert und die Verschlüsselung von Daten im Ruhezustand ist aktiviert.
- Die lokale Festplattenverschlüsselung ist AWS KMS als Schlüsselanbieter aktiviert und die Amazon S3 S3-Verschlüsselung ist deaktiviert.
- Die EBS-Verschlüsselung ist aktiviert.

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-configuration '{
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": false,
    "EnableAtRestEncryption": true,
    "AtRestEncryptionConfiguration": {
      "LocalDiskEncryptionConfiguration": {
        "EnableEbsEncryption": true,
        "EncryptionKeyProviderType": "AwsKms",
        "AwsKmsKey": "arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      }
    }
  }
}'
```

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

SSE-EMR-WAL wird für die EMR-WAL-Verschlüsselung verwendet

```
aws emr create-security-configuration --name "MySecConfig" \
  --security-configuration '{
    "EncryptionConfiguration": {
      "EMRWALEncryptionConfiguration":{ },
      "EnableInTransitEncryption":false, "EnableAtRestEncryption":false
    }
  }'
```

EnableInTransitEncryption und könnte EnableAtRestEncryption immer noch wahr sein, wenn Sie die entsprechende Verschlüsselung aktivieren möchten.

Das nachstehende Beispiel veranschaulicht das folgende Szenario:

- SSE-KMS-WAL wird für die EMR-WAL-Verschlüsselung verwendet
- Serverseitige Verschlüsselung wird AWS Key Management Service als Schlüsselanbieter verwendet

```
aws emr create-security-configuration --name "MySecConfig" \
  --security-configuration '{
    "EncryptionConfiguration": {
      "EMRWALEncryptionConfiguration":{
        "AwsKmsKey":"arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
      },
      "EnableInTransitEncryption":false, "EnableAtRestEncryption":false
    }
  }'
```

EnableInTransitEncryption und könnte EnableAtRestEncryption immer noch wahr sein, wenn Sie die entsprechende Verschlüsselung aktivieren möchten.

JSON-Referenz für Verschlüsselungseinstellungen

In der folgenden Tabelle finden Sie die JSON-Parameter für die Verschlüsselungseinstellungen sowie eine Beschreibung der zulässigen Werte für die einzelnen Parameter.

Parameter	Beschreibung
<code>"EnableInTransitEncryption" : true false</code>	Geben Sie <code>true</code> an, um die Verschlüsselung der Daten während der Übertragung zu aktivieren, und <code>false</code> , um sie zu deaktivieren. Wenn der Parameter nicht definiert wird, gilt <code>false</code> und die Verschlüsselung von Daten während der Übertragung ist deaktiviert.
<code>"EnableAtRestEncryption": true false</code>	Geben Sie <code>true</code> an, um die Verschlüsselung der Daten im Ruhezustand zu aktivieren, und <code>false</code> , um sie zu deaktivieren. Wenn der Parameter nicht definiert wird, gilt <code>false</code> und die Verschlüsselung von Daten im Ruhezustand ist deaktiviert.
Parameter für die Verschlüsselung während der Übertragung	
<code>"InTransitEncryptionConfiguration" :</code>	Gibt eine Sammlung von Werten für die Verschlüsselung von Daten während der Übertragung an, wenn <code>EnableInTransitEncryption true</code> ist.
<code>"CertificateProviderType": "PEM" "Custom"</code>	Gibt an, ob PEM-Zertifikate verwendet werden sollen, auf die über eine ZIP-Datei oder über einen Custom Zertifikatsanbieter verwiesen wird. Wenn angegeben, <code>S3Object</code> muss PEM es sich um einen Verweis auf den Speicherort einer ZIP-Datei mit den Zertifikaten in Amazon S3 handeln. Wenn <code>Custom</code> angegeben ist, <code>S3Object</code> muss es sich um einen Verweis auf den Speicherort einer JAR-Datei in Amazon S3 handeln, gefolgt von einem <code>CertificateProviderClass</code> Eintrag.
<code>"S3Object" : " <i>ZipLocation</i> " " <i>JarLocation</i> "</code>	Stellt den Speicherort in Amazon S3 für eine ZIP-Datei bereit, wenn PEM angegeben, oder für eine JAR-Datei, wenn Custom angegeben

Parameter	Beschreibung
	. Beim Format kann es sich um einen Pfad (beispielsweise <code>s3://MyConfig/artifacts/CertFiles.zip</code>) oder einen ARN (beispielsweise <code>arn:aws:s3:::Code/MyCertProvider.jar</code>) handeln. Wenn eine ZIP-Datei ausgewählt wurde, muss sie Dateien enthalten, deren Namen <code>privateKey.pem</code> und <code>certificateChain.pem</code> sind. Eine Datei mit dem Namen <code>trustedCertificates.pem</code> ist optional.
"CertificateProviderClass" : " <i>MyClassID</i> "	Nur erforderlich, wenn für angegeben Custom istCertificateProviderType . <i>MyClassID</i> gibt einen vollständigen Klassennamen an, der in der JAR-Datei deklariert ist, die die TLSArtifacts Provider-Schnittstelle implementiert. Beispiel, <code>com.mycompany.MyCertProvider</code> .
Parameter für die Verschlüsselung im Ruhezustand	
"AtRestEncryptionConfiguration" :	Gibt eine Sammlung von Werten für die Verschlüsselung im Ruhezustand an, wenn dies der EnableAtRestEncryption Fall isttrue, einschließlich Amazon S3 S3-Verschlüsselung und Verschlüsselung lokaler Festplatten.
Amazon S3 S3-Verschlüsselungsparameter	
"S3EncryptionConfiguration" :	Gibt eine Sammlung von Werten an, die für die Amazon S3 S3-Verschlüsselung mit dem Amazon EMR File System (EMRFS) verwendet werden.

Parameter	Beschreibung
"EncryptionMode" : "SSE-S3" "SSE-KMS" "CSE-KMS" "CSE-Custom"	Gibt den Typ der zu verwendenden Amazon S3 S3-Verschlüsselung an. Wenn SSE-S3 angegeben, sind keine weiteren Amazon S3 S3-Verschlüsselungswerte erforderlich. Wenn entweder SSE-KMS oder angegeben CSE-KMS ist, muss ein AWS KMS key ARN als <code>AwsKmsKey</code> Wert angegeben werden. Wenn CSE-Custom ausgewählt wurde, müssen <code>S3Object-</code> und <code>EncryptionKeyProviderClass</code> -Werte angegeben werden.
"AwsKmsKey" : " <i>MyKeyARN</i> "	Nur erforderlich, wenn SSE-KMS oder CSE-KMS für <code>EncryptionMode</code> angegeben wurden. <i>MyKeyARN</i> muss ein vollständig angegebener ARN für einen Schlüssel sein (z. B. <code>arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012</code>).
"S3Object" : " <i>JarLocation</i> "	Nur erforderlich, wenn für angegeben CSE-Custom <code>isCertificateProviderType</code> . <i>JarLocation</i> gibt den Speicherort in Amazon S3 für eine JAR-Datei an. Beim Format kann es sich um einen Pfad (beispielsweise <code>s3://MyConfig/artifacts/MyKeyProvider.jar</code>) oder einen ARN (beispielsweise <code>arn:aws:s3:::Code/MyKeyProvider.jar</code>) handeln.

Parameter	Beschreibung
<code>"EncryptionKeyProviderClass" : "MyS3KeyClassID "</code>	Nur erforderlich, wenn für angegeben CSE-Custom istEncryptionMode . <i>MyS3KeyClassID</i> gibt den vollständigen Klassennamen einer Klasse an, die in der Anwendung deklariert ist, die die EncryptionMaterial sProvider Schnittstelle implementiert; zum Beispiel <i>com.mycompany.MyS3KeyProvider</i> .
Parameter für Verschlüsselung auf dem lokalen Datenträger	
<code>"LocalDiskEncryptionConfiguration"</code>	Gibt die Schlüsselanbieter und die entsprechenden Werte an, die für die lokale Laufwerksverschlüsselung verwendet werden müssen.
<code>"EnableEbsEncryption": true false</code>	Geben Sie <code>true</code> an, ob die EBS-Verschlüsselung aktiviert werden soll. Die EBS-Verschlüsselung verschlüsselt das EBS-Root-Geräte-Volumen und die angeschlossenen Speichervolumen. Um die EBS-Verschlüsselung zu verwenden, müssen Sie als <code>AwsKms EncryptionKeyProviderType</code> angegeben werden.
<code>"EncryptionKeyProviderType": "AwsKms" "Custom"</code>	Gibt den Schlüsselanbieter an. Wenn <code>AwsKms</code> angegeben, muss ein KMS-Schlüssel-ARN als <code>AwsKmsKey</code> Wert angegeben werden. Wenn Custom ausgewählt wurde, müssen <code>S3Object-</code> und <code>EncryptionKeyProviderClass</code> -Werte angegeben werden.
<code>"AwsKmsKey : " MyKeyARN"</code>	Nur erforderlich, wenn für angegeben <code>AwsKms</code> istType. <i>MyKeyARN</i> muss ein vollständig spezifizierter ARN für einen Schlüssel sein (z. B. <code>arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-456789012123</code>).

Parameter	Beschreibung
"S3Object" : " <i>JarLocation</i> "	Nur erforderlich, wenn für angegeben Custom <code>isCertificateProviderType</code> . <i>JarLocation</i> gibt den Speicherort in Amazon S3 für eine JAR-Datei an. Beim Format kann es sich um einen Pfad (beispielsweise <code>s3://MyConfig/artifacts/MyKeyProvider.jar</code>) oder einen ARN (beispielsweise <code>arn:aws:s3:::Code/MyKeyProvider.jar</code>) handeln.
"EncryptionKeyProviderClass" : " <i>MyLocalDiskKeyClassID</i> "	Nur erforderlich, wenn für angegeben Custom <code>isType</code> . <i>MyLocalDiskKeyClassID</i> gibt den vollständigen Klassennamen einer Klasse an, die in der Anwendung deklariert ist, die die <code>EncryptionMaterialsProvider</code> Schnittstelle implementiert; zum Beispiel <i>com.mycompany.MyLocalDiskKeyProvider</i> .
EMR WAL-Verschlüsselungsparameter	
"EMRWALEncryptionConfiguration"	Gibt den Wert für die EMR-WAL-Verschlüsselung an.
"AwsKmsKey"	Gibt die CMK-Schlüssel-ID Arn an.

Konfiguration der Kerberos-Authentifizierung

Eine Sicherheitskonfiguration mit Kerberos-Einstellungen kann nur von einem Cluster verwendet werden, das mit Kerberos-Attributen erstellt wurde, andernfalls tritt ein Fehler auf. Weitere Informationen finden Sie unter [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#). Kerberos ist nur in Amazon-EMR-Version 5.10.0 und höher verfügbar.

Kerberos-Einstellungen unter Verwendung der Konsole angeben

Wählen Sie anhand der folgenden Anleitungen Optionen in Kerberos authentication (Kerberos-Authentifizierung) aus.

Parameter		Beschreibung
Kerberos		Gibt an, dass Kerberos für Cluster aktiviert ist, die diese Sicherheitskonfiguration verwenden. Wenn ein Cluster diese Sicherheitskonfiguration verwendet, müssen für den Cluster auch Kerberos-Einstellungen angegeben sein, andernfalls tritt ein Fehler auf.
Anbieter	Cluster-dediziertes KDC	Gibt an, dass Amazon EMR einen KDC auf dem Primärknoten eines Clusters erstellt, der diese Sicherheitskonfiguration verwendet. Sie geben den Bereichsnamen und das KDC-Administratorkennwort an, wenn Sie den Cluster erstellen. Bei Bedarf können Sie von anderen Clustern aus auf diesen KDC verweisen. Erstellen Sie diese Cluster mit einer anderen Sicherheitskonfiguration, geben Sie ein externes KDC an und verwenden Sie den Bereichsnamen und das KDC-Administratorkennwort, die Sie für das clusterspezifische KDC angeben.
	Externes KDC	Nur in Amazon-EMR-Version 5.20.0 und höher verfügbar. Gibt an, dass Cluster, die diese Sicherheitskonfiguration verwenden, Kerberos-Prinzipale mithilfe eines KDC-Servers außerhalb des Clusters authentifizieren. Auf dem Cluster wird kein KDC erstellt. Sie geben den Bereichsnamen und das KDC-Administratorkennwort an, wenn Sie den Cluster erstellen.
Gültigkeitsdauer des Tickets		Optional. Gibt den Zeitraum an, für den ein vom KDC ausgestelltes Kerberos-Ticket auf Clustern gültig ist, die diese Sicherheitskonfiguration verwenden. Ticket-Gültigkeitsdauern werden aus Sicherheitsgründen beschränkt. Cluster-Anwendungen und Services verlängern Tickets automatisch, wenn sie ablaufen. Benutzer, die eine Verbindung mit dem

Parameter		Beschreibung
		Cluster über SSH mit Kerberos-Anmeldeinformationen einrichten, müssen <code>kinit</code> von der Befehlszeile des Primärknoten aus ausführen, um eine Verlängerung auszuführen, nachdem ein Ticket abgelaufen ist.
Bereichsübergreifende Vertrauensstellung		Gibt eine bereichsübergreifende Vertrauensstellung zwischen einem clusterspezifischen KDC auf Clustern, die diese Sicherheitskonfiguration verwenden, und einem KDC in einem anderen Kerberos-Bereich an. Prinzipale (in der Regel Benutzer) aus einem anderen Bereich werden gegenüber Clustern authentifiziert, die diese Konfiguration verwenden. Eine zusätzliche Konfiguration im anderen Kerberos-Bereich ist erforderlich. Weitere Informationen finden Sie unter Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain.
Realitätsübergreifende Vertrauensstellungen	Bereich	Gibt den Kerberos-Bereichsnamen des anderen Bereichs in der Vertrauensstellung an. Gemäß der Konvention sind Kerberos-Bereichsnamen mit dem Domainnamen identisch, jedoch ausschließlich in Großbuchstaben.
	Domain	Gibt den Domain-Namen des anderen Bereichs in der Vertrauensstellung an.

Parameter		Beschreibung
	Admin-Server	Gibt den Fully Qualified Domain Name (FQDN, vollständig qualifizierter Domainname) oder IP-Adresse des Admin-Servers im anderen Bereich der Vertrauensstellung an. Der Admin-Server und KDC-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, kommunizieren jedoch über andere Ports. Falls kein Port angegeben ist, wird Port 749 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :749</code>).
	KDC-Server	Gibt den vollständig qualifizierten Domain-Namen (FQDN, Fully Qualified Domain Name) oder IP-Adresse des KDC-Servers im anderen Bereich der Vertrauensstellung an. Der KDC-Server und Admin-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, nutzen jedoch andere Ports. Falls kein Port angegeben ist, wird Port 88 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :88</code>).
Externes KDC		Gibt an, dass das externe KDC des Clusters vom Cluster verwendet wird.

Parameter		Beschreibung	
Eigenschaften des externen KDCs	Admin-Server	Gibt den vollqualifizierten Domainnamen oder die IP-Adresse des externen Admin-Servers an. Der Admin-Server und KDC-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, kommunizieren jedoch über andere Ports. Falls kein Port angegeben ist, wird Port 749 verwendet , da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :749</code>).	
	KDC-Server	Gibt den vollqualifizierten Domainnamen des externen KDC-Servers an. Der KDC-Server und Admin-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, nutzen jedoch andere Ports. Falls kein Port angegeben ist, wird Port 88 verwendet , da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :88</code>).	
	Active-Directory-Integration		Gibt an, dass die Kerberos-Prinzipalauthentifizierung in eine Microsoft-Active-Directory-Domain integriert ist.
	Active-Directory-Integrationseigenschaften	Active-Directory-Bereich	Gibt den Kerberos-Bereichsnamen der Active-Directory-Domain an. Gemäß der Konvention sind Kerberos-Bereichsnamen in der Regel identisch mit dem Domainnamen, jedoch ausschließlich in Großbuchstaben.
Active-Directory-Domain		Gibt den Active-Directory-Domainnamen an.	

Parameter			Beschreibung
		Active-Directory-Server	Gibt den vollqualifizierten Domainnamen des Microsoft Active Directory-Domain-Controllers an.

Angeben von Kerberos-Einstellungen mit dem AWS CLI

Die folgende Referenztabelle zeigt JSON-Parameter für Kerberos-Einstellungen in einer Sicherheitskonfiguration. Beispielkonfigurationen finden Sie unter [Beispiele für Konfigurationen](#).

Parameter		Beschreibung
<code>"AuthenticationConfiguration": {</code>		Erforderlich für Kerberos. Gibt an, dass eine Authentifizierungskonfiguration Teil dieser Sicherheitskonfiguration ist.
	<code>"KerberosConfiguration": {</code>	Erforderlich für Kerberos. Gibt die Kerberos-Konfigurationseigenschaften an.
	<code>"Provider": "ClusterDedicatedKdc",</code> –oder– <code>"Provider": "ExternalKdc",</code>	<i>ClusterDedicatedKdc</i> gibt an, dass Amazon EMR einen KDC auf dem Primärknoten eines Clusters erstellt, der diese Sicherheitskonfiguration verwendet. Sie geben den Bereichsnamen und das KDC-Administrator Kennwort an, wenn Sie den Cluster erstellen. Bei Bedarf können Sie von anderen Clustern aus auf diesen KDC verweisen. Erstellen Sie diese Cluster mit einer anderen Sicherheitskonfiguration, geben Sie einen externen KDC an und verwenden Sie den Bereichsnamen und das KDC-Administrator Kennwort,

Parameter	Beschreibung
	die Sie beim Erstellen des Clusters mit dem Cluster-dedizierten KDC angegeben haben. <i>ExternalKdc</i> gibt an, dass der Cluster ein externes KDC verwendet. Amazon EMR erstellt kein KDC auf dem Primärknoten. Ein Cluster, der diese Sicherheitskonfiguration verwendet, muss den Bereichsnamen und das KDC-Administrator Kennwort des externen KDC angeben.
<pre>"ClusterDedicatedKdcConfiguration": { "TicketLifetimeInHours": 24,</pre>	Erforderlich, wenn <i>ClusterDedicatedKdc</i> angegeben ist. Optional. Gibt den Zeitraum an, für den ein vom KDC ausgestelltes Kerberos-Ticket auf Clustern gültig ist, die diese Sicherheitskonfiguration verwenden. Ticket-Gültigkeitsdauern werden aus Sicherheitsgründen beschränkt. Cluster-Anwendungen und Services verlängern Tickets automatisch, wenn sie ablaufen. Benutzer, die eine Verbindung mit dem Cluster über SSH mit Kerberos-Anmeldeinformationen einrichten, müssen <code>kinit</code> von der Befehlszeile des Primärknoten aus ausführen, um eine Verlängerung auszuführen, nachdem ein Ticket abgelaufen ist.

Parameter	Beschreibung
<pre>"CrossRealmTrustConfiguration": {</pre>	Gibt eine bereichsübergreifende Vertrauensstellung zwischen einem clusterspezifischen KDC auf Clustern, die diese Sicherheitskonfiguration verwenden, und einem KDC in einem anderen Kerberos-Bereich an. Prinzipale (in der Regel Benutzer) aus einem anderen Bereich werden gegenüber Clustern authentifiziert, die diese Konfiguration verwenden. Eine zusätzliche Konfiguration im anderen Kerberos-Bereich ist erforderlich. Weitere Informationen finden Sie unter Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain.
<pre> "Realm": "<i>KDC2.COM</i>",</pre>	Gibt den Kerberos-Bereichsnamen des anderen Bereichs in der Vertrauensstellung an. Gemäß der Konvention sind Kerberos-Bereichsnamen mit dem Domainnamen identisch, jedoch ausschließlich in Großbuchstaben.
<pre> "Domain": "<i>kd2.com</i>",</pre>	Gibt den Domain-Namen des anderen Bereichs in der Vertrauensstellung an.

Parameter	Beschreibung
<pre>"AdminServer": "kdc.com:749 ",</pre>	Gibt den Fully Qualified Domain Name (FQDN, vollständig qualifizierter Domainname) oder IP-Adresse des Admin-Servers im anderen Bereich der Vertrauensstellung an. Der Admin-Server und KDC-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, kommunizieren jedoch über andere Ports. Falls kein Port angegeben ist, wird Port 749 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :749</code>).
<pre>"KdcServer": "kdc.com:88 "</pre>	Gibt den vollständig qualifizierten Domain-Namen (FQDN, Fully Qualified Domain Name) oder IP-Adresse des KDC-Servers im anderen Bereich der Vertrauensstellung an. Der KDC-Server und Admin-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, nutzen jedoch andere Ports. Falls kein Port angegeben ist, wird Port 88 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :88</code>).

Parameter	Beschreibung
}	
}	
"ExternalKdcConfiguration": {	Erforderlich, wenn <i>ExternalKdc</i> angegeben ist.
"TicketLifetimeInHours": 24,	Optional. Gibt den Zeitraum an, für den ein vom KDC ausgestelltes Kerberos-Ticket auf Clustern gültig ist, die diese Sicherheitskonfiguration verwenden. Ticket-Gültigkeitsdauern werden aus Sicherheitsgründen beschränkt. Cluster-Anwendungen und Services verlängern Tickets automatisch, wenn sie ablaufen. Benutzer, die eine Verbindung mit dem Cluster über SSH mit Kerberos-Anmeldeinformationen einrichten, müssen <code>kinit</code> von der Befehlszeile des Primärknoten aus ausführen, um eine Verlängerung auszuführen, nachdem ein Ticket abgelaufen ist.
"KdcServerType": "Single",	Gibt an, dass auf einen einzelnen KDC-Server verwiesen wird. <code>Single</code> ist derzeit der einzige unterstützte Wert.

Parameter			Beschreibung
		"AdminServer": <i>"kdc.com:749 "</i> ,	Gibt den vollqualifizierten Domainnamen oder die IP-Adresse des externen Admin-Servers an. Der Admin-Server und KDC-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, kommunizieren jedoch über andere Ports. Falls kein Port angegeben ist, wird Port 749 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :749</code>).
		"KdcServer": <i>"kdc.com:88 "</i> ,	Gibt den vollqualifizierten Domainnamen des externen KDC-Servers an. Der KDC-Server und Admin-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, nutzen jedoch andere Ports. Falls kein Port angegeben ist, wird Port 88 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :88</code>).
		"AdIntegrationConfiguration": {	Gibt an, dass die Kerberos-Prinzipalauthentifizierung in eine Microsoft-Active-Directory-Domain integriert ist.

Parameter		Beschreibung
	<pre>"AdRealm": "AD.DOMAIN .COM ",</pre>	Gibt den Kerberos-Bereichsnamen der Active-Directory-Domain an. Gemäß der Konvention sind Kerberos-Bereichsnamen in der Regel identisch mit dem Domainnamen, jedoch ausschließlich in Großbuchstaben.
	<pre>"AdDomain": "ad.domain .com "</pre>	Gibt den Active-Directory-Domainnamen an.
	<pre>"AdServer": "ad.domain .com "</pre>	Gibt den vollqualifizierten Domainnamen des Microsoft Active Directory-Domain-Controllers an.
	}	
	}	
	}	
	}	

Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3

Mit IAM-Rollen für EMRFS können Sie unterschiedliche Berechtigungen für EMRFS-Daten in Amazon S3 bereitstellen. Sie erstellen Rollenzuordnungen, die eine IAM-Rolle spezifizieren, die für Berechtigungen verwendet wird, wenn eine Zugriffsanforderung eine von Ihnen angegebene Kennung enthält. Bei der ID kann es sich um einen Hadoop-Benutzer oder eine Hadoop-Rolle oder ein Amazon-S3-Präfix handeln.

Weitere Informationen finden Sie unter [Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3](#).

Angeben von IAM-Rollen für EMRFS mithilfe von AWS CLI

Im Folgenden finden Sie ein JSON-Beispiel für die Angabe benutzerdefinierter IAM-Rollen für EMRFS innerhalb einer Sicherheitskonfiguration. Es zeigt Rollenzuordnungen für die drei verschiedenen Identifier-Typen, gefolgt von einer Parameterreferenz.

```
{
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_demo_s3_buckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://amzn-s3-demo-bucket1/", "s3://amzn-s3-demo-bucket2/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

Parameter	Beschreibung
"AuthorizationConfiguration":	Erforderlich
"EmrFsConfiguration":	Erforderlich Enthält Rollenzuordnungen.
"RoleMappings":	Erforderlich Enthält eine oder mehrere Rollenzuordnungsdefinitionen. Rollenzuordnungen werden in der Reihenfolge bewertet, in der sie von oben nach unten angezeigt werden. Wenn eine Rollenzuweisung für einen EMRFS-Datenaufruf in Amazon S3 als wahr bewertet wird, werden keine weiteren Rollenzuordnungen ausgewertet und EMRFS

Parameter	Beschreibung
	verwendet die angegebene IAM-Rolle für die Anfrage. Rollenzuordnungen bestehen aus den folgenden erforderlichen Parametern:
"Role":	Gibt den ARN-Bezeichner einer IAM-Rolle im Format <code>arn:aws:iam:: <i>account-id</i> :role/<i>role-name</i></code> an. Dies ist die IAM-Rolle, die Amazon EMR übernimmt, wenn die EMRFS-Anfrage an Amazon S3 mit einer der angegebenen Identifizierer übereinstimmt.
"IdentifierType":	Kann einer der folgenden sein: "User" gibt an, dass es sich bei den Kennungen um einen oder mehrere Hadoop-Benutzer handelt, bei denen es sich um Linux-Kontobenutzer oder Kerberos-Prinzipale handeln kann. Wenn die EMRFS-Anfrage von dem oder den angegebenen Benutzern stammt, wird die IAM-Rolle übernommen. "Prefix" gibt an, dass der Identifizierer ein Amazon-S3-Speicherort ist. Die IAM-Rolle wird für Anrufe an den Standort oder die Standorte mit den angegebenen Präfixen übernommen. Das Präfix <code>s3://amzn-s3-demo-bucket/</code> entspricht beispielsweise <code>s3://amzn-s3-demo-bucket/mydir</code> und <code>s3://amzn-s3-demo-bucket/yetanotherdir</code>. "Group" gibt an, dass es sich bei den Identifikatoren um eine oder mehrere Hadoop-Gruppen handelt. Die IAM-Rolle wird übernommen, wenn die Anfrage von einem Benutzer in der oder den angegebenen Gruppen stammt.

Parameter	Beschreibung
"Identifiers":	Gibt einen oder mehrere Kennungen des entsprechenden Kennungstyps an. Trennen Sie mehrere Bezeichner durch Kommas ohne Leerzeichen.

Metadaten-Serviceanfragen an EC2 Amazon-Instances konfigurieren

Instance-Metadaten sind Daten über eine Instance, mit denen Sie die ausgeführte Instance konfigurieren und verwalten können. Sie können mit einer der folgenden Methoden auf Instance-Metadaten aus einer laufenden Instance zugreifen:

- Instance Metadata Service Version 1 (IMDSv1) — eine Anforderungs-/Antwortmethode
- Instanz-Metadatendienst Version 2 (IMDSv2) — eine sitzungsorientierte Methode

Während Amazon IMDSv1 sowohl als auch EC2 unterstützt IMDSv2, unterstützt Amazon EMR IMDSv2 in Amazon EMR 5.23.1, 5.27.1, 5.32 oder höher und 6.2 oder höher. In diesen Versionen werden Amazon EMR-Komponenten IMDSv2 für alle IMDS-Aufrufe verwendet. Für IMDS-Aufrufe in Ihrem Anwendungscode können Sie sowohl als auch IMDSv1 verwenden oder das IMDS so konfigurieren IMDSv2, dass es nur IMDSv2 für zusätzliche Sicherheit verwendet wird. Wenn Sie angeben, dass dies verwendet werden IMDSv2 muss, funktioniert IMDSv1 es nicht mehr.

Weitere Informationen finden [Sie unter Konfiguration des Instance-Metadaten-Service](#) im EC2 Amazon-Benutzerhandbuch.

Note

In früheren Versionen von Amazon EMR 5.x oder 6.x IMDSv1 führt das Ausschalten zu einem Cluster-Startfehler, da Amazon EMR-Komponenten IMDSv1 für alle IMDS-Aufrufe verwendet werden. Stellen Sie beim Ausschalten sicher IMDSv1, dass jede benutzerdefinierte Software, die verwendet wird, auf aktualisiert ist. IMDSv1 IMDSv2

Spezifizieren Sie die Konfiguration des Instance Metadata Services mit dem AWS CLI

Im Folgenden finden Sie ein JSON-Beispiel für die Angabe des Amazon EC2 Instance Metadata Service (IMDS) innerhalb einer Sicherheitskonfiguration. Die Verwendung einer benutzerdefinierten Sicherheitskonfiguration ist optional.

```
{
  "InstanceMetadataServiceConfiguration" : {
    "MinimumInstanceMetadataServiceVersion": integer,
    "HttpPutResponseHopLimit": integer
  }
}
```

Parameter	Beschreibung
"InstanceMetadataServiceConfiguration":	Wenn Sie IMDS nicht innerhalb einer Sicherheitskonfiguration angeben und eine Amazon EMR-Version verwenden, die dies erfordert IMDSv1, verwendet Amazon EMR standardmäßig IMDSv1 als Mindestversion des Instance-Metadatendienstes. Wenn Sie Ihre eigene Konfiguration verwenden möchten, sind die beiden folgenden Parameter erforderlich.
"MinimumInstanceMetadataServiceVersion":	Erforderlich Geben Sie 1 oder 2 an. Der Wert 1 erlaubt IMDSv1 und IMDSv2. Der Wert 2 erlaubt nur IMDSv2.
"HttpPutResponseHopLimit":	Erforderlich Das gewünschte HTTP PUT-Antwort-Hop-Limit für Instance-Metadatenanfragen. Je größer die Zahl ist, desto weiter können sich die Instance-Metadatenanfragen bewegen. Standard: 1. Einen Ganzzahlwert von 1 bis 64 angeben.

Die Konfiguration des Instance Metadata Services mit der Konsole angeben

Sie können die Verwendung von IMDS für einen Cluster konfigurieren, wenn Sie ihn von der Amazon-EMR-Konsole aus starten.

So konfigurieren Sie die Verwendung von IMDS mithilfe der Konsole:

1. Wenn Sie auf der Seite Sicherheitskonfigurationen eine neue Sicherheitskonfiguration erstellen, wählen Sie unter der Einstellung EC2Instanz-Metadatendienst die Option EC2Instanz-Metadatendienst konfigurieren aus. Diese Konfiguration wird nur in Amazon EMR 5.23.1, 5.27.1, 5.32 oder höher und 6.2 oder höher unterstützt.
2. Für Minimum Instance Metadata Service Version wählen Sie eine der folgenden Optionen aus:
 - Schalten Sie die IMDSv1 Option aus und lassen Sie sie nur zu IMDSv2, wenn Sie nur IMDSv2 auf diesem Cluster zulassen möchten. Weitere Informationen finden Sie [unter Umstellung auf die Nutzung des Instance-Metadaten-Service Version 2](#) im EC2 Amazon-Benutzerhandbuch.
 - Erlauben Sie beides IMDSv1 und IMDSv2 auf dem Cluster, wenn Sie IMDSv2 auf diesem Cluster eine sitzungsorientierte Einstellung zulassen IMDSv1 möchten.
3. Denn IMDSv2 Sie können auch die zulässige Anzahl von Netzwerk-Hops für das Metadaten-Token konfigurieren, indem Sie das HTTP-Put-Response-Hop-Limit auf eine Ganzzahl zwischen 1 und 64 festlegen.

Weitere Informationen finden Sie [unter Konfiguration des Instance-Metadaten-Service](#) im EC2 Amazon-Benutzerhandbuch.

Weitere Informationen finden Sie [unter Instance-Details konfigurieren und Instance-Metadaten-Service](#) konfigurieren im EC2 Amazon-Benutzerhandbuch.

Geben Sie eine Sicherheitskonfiguration für einen Amazon EMR-Cluster an

Sie können bei der Erstellung eines Clusters die Verschlüsselungseinstellungen festlegen, indem Sie eine Sicherheitskonfiguration angeben. Sie können das AWS Management Console oder das AWS CLI verwenden.

Console

Um eine Sicherheitskonfiguration mit der Konsole anzugeben

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Suchen Sie unter Sicherheitskonfiguration und Berechtigungen das Feld Sicherheitskonfiguration. Wählen Sie das Dropdownmenü oder klicken Sie auf Durchsuchen, um den Namen einer Sicherheitskonfiguration auszuwählen, die Sie zuvor erstellt haben. Wählen Sie alternativ VPC erstellen, um eine VPC zu erstellen, die Sie für Ihren Cluster verwenden können.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

CLI

Um eine Sicherheitskonfiguration anzugeben mit dem AWS CLI

- Verwenden Sie `aws emr create-cluster` und Sie können mithilfe von `--security-configuration` *MySecConfig* wahlweise eine Sicherheitskonfiguration verwenden, wobei *MySecConfig* der Name der Sicherheitskonfiguration ist, wie im folgenden Beispiel dargestellt. Der angegebene `--release-label` muss 4.8.0 oder höher sein und `--instance-type` kann als jeder verfügbare Typ ausgewählt werden.

```
aws emr create-cluster --instance-type m5.xlarge --release-label emr-5.0.0 --  
security-configuration mySecConfig
```

Datenschutz bei Amazon EMR

Das [Modell der AWS gemeinsamen Verantwortung](#) gilt für den Datenschutz in Amazon EMR. AWS ist, wie in diesem Modell beschrieben, für den Schutz der globalen Infrastruktur verantwortlich, auf der die gesamte AWS Cloud läuft. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt umfasst die Sicherheitskonfiguration und die Verwaltungsaufgaben für die AWS, die Sie verwenden. Weitere Informationen zum Datenschutz

finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im [Amazon Shared Responsibility-Modell](#) und im [DSGVO-Blogbeitrag](#) auf dem AWS Security Blog.

Aus Datenschutzgründen empfehlen wir Ihnen, Ihre AWS Kontoanmeldeinformationen zu schützen und individuelle Konten bei einzurichten AWS Identity and Access Management. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie TLS, um mit AWS Ressourcen zu kommunizieren. Wir erfordern TLS 1.2.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen innerhalb der AWS Dienste.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu sichern.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern wie z. B. im Feld Name keine sensiblen, identifizierenden Informationen wie Kontonummern von Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon EMR oder anderen AWS Services über die Konsole AWS CLI, API oder AWS SDKs arbeiten. Alle Daten, die Sie in Amazon EMR oder andere Services eingeben, werden möglicherweise in Diagnoseprotokolle aufgenommen. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Verschlüsseln Sie Daten im Ruhezustand und bei der Übertragung mit Amazon EMR

Die Datenverschlüsselung verhindert, dass nicht autorisierte Benutzer Daten auf einem Cluster und in den dazugehörigen Datenspeichersystemen lesen können. Dies gilt für auf persistenten Medien gespeicherte Daten, auch als Daten im Ruhezustand bezeichnet, und für Daten, die während

der Übertragung im Netzwerk möglicherweise abgefangen werden, auch als Daten während der Übertragung bezeichnet.

Ab Amazon-EMR-Version 4.8.0 können Sie mit Amazon-EMR-Sicherheitskonfigurationen Datenverschlüsselungseinstellungen für Cluster einfacher konfigurieren. Sicherheitskonfigurationen stellen Einstellungen bereit, um die Sicherheit von Daten während der Übertragung und im Ruhezustand in Amazon Elastic Block Store (Amazon EBS)-Volumen und EMRFS in Amazon S3 zu unterstützen.

Optional können Sie ab Amazon EMR Version 4.1.0 und höher eine transparente Verschlüsselung in HDFS konfigurieren, die nicht unter Verwendung von Sicherheitskonfigurationen konfiguriert ist. Weitere Informationen finden Sie unter [Transparente Verschlüsselung in HDFS in Amazon EMR](#) in Amazon-EMR-Versionenhinweise.

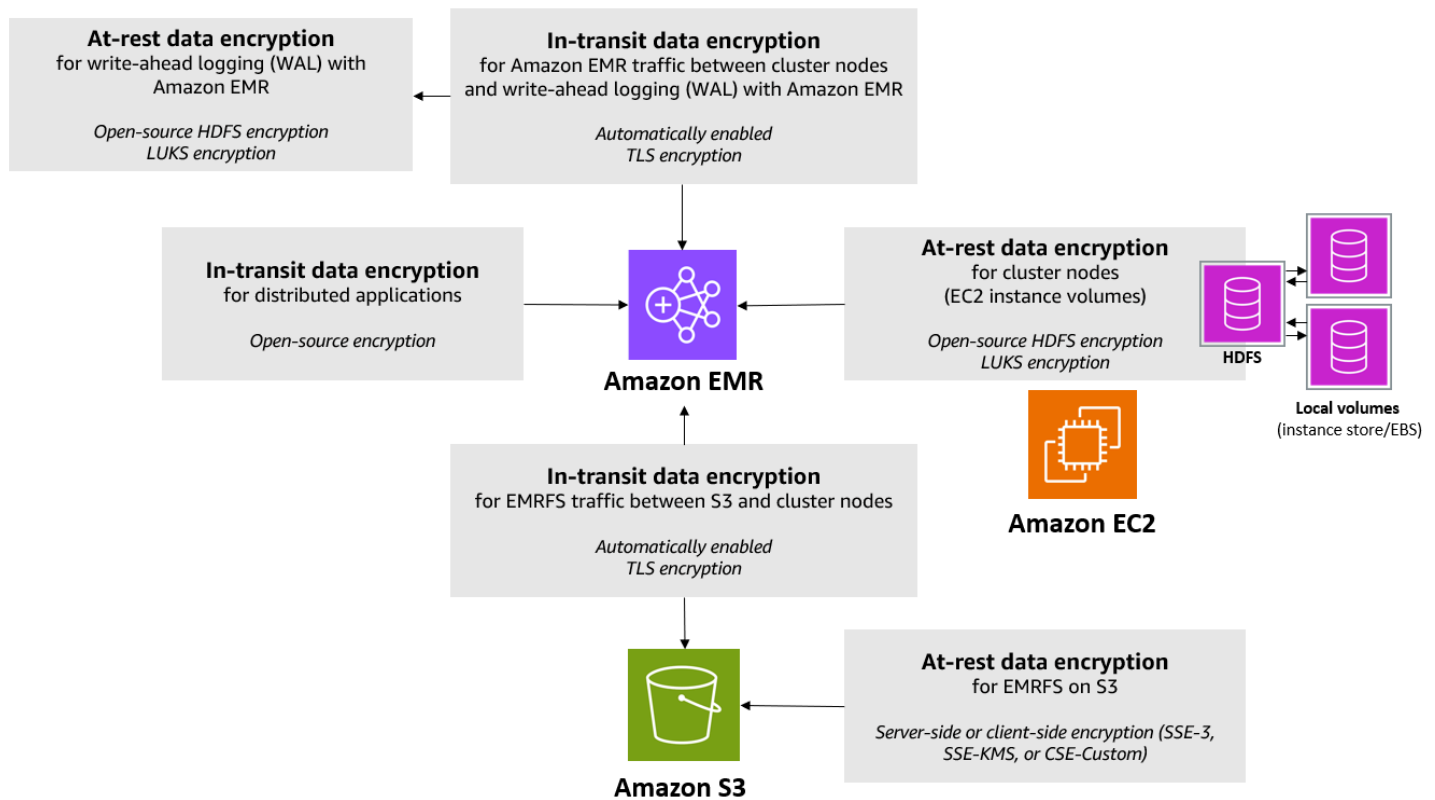
Themen

- [Verschlüsselungsoptionen für Amazon EMR](#)
- [Verschlüsselung im Ruhezustand mithilfe eines Kunden-KMS-Schlüssels für den EMR WAL-Service](#)
- [Schlüssel und Zertifikate für die Datenverschlüsselung mit Amazon EMR erstellen](#)
- [Grundlegendes zur Verschlüsselung bei der Übertragung](#)

Verschlüsselungsoptionen für Amazon EMR

Mit Amazon EMR-Versionen 4.8.0 und höher können Sie eine Sicherheitskonfiguration verwenden, um Einstellungen für die Verschlüsselung von Daten im Ruhezustand, Daten während der Übertragung oder beidem festzulegen. Wenn Sie die Datenverschlüsselung im Ruhezustand aktivieren, können Sie wählen, ob Sie EMRFS-Daten in Amazon S3, Daten auf lokalen Festplatten oder beide verschlüsseln möchten. Jede von Ihnen erstellte Sicherheitskonfiguration wird in Amazon EMR und nicht in der Cluster-Konfiguration gespeichert. Daher können Sie die Konfiguration bei jeder Cluster-Erstellung problemlos wiederverwenden, um die Datenverschlüsselung zu konfigurieren. Weitere Informationen finden Sie unter [Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI](#).

Das folgende Diagramm zeigt die verschiedenen Datenverschlüsselungsoptionen, die für die Sicherheitskonfigurationen zur Verfügung stehen.



Die folgenden Verschlüsselungsoptionen stehen ebenfalls zur Verfügung und werden nicht mit einer Sicherheitskonfiguration konfiguriert:

- Optional können Sie mit Amazon-EMR-Versionen 4.1.0 und höher eine transparente Verschlüsselung in HDFS konfigurieren. Weitere Informationen finden Sie unter [Transparente Verschlüsselung in HDFS in Amazon EMR](#) in Amazon-EMR-Versionenhinweise.
- Wenn Sie eine Version von Amazon EMR verwenden, die Sicherheitskonfigurationen nicht unterstützt, können Sie die Verschlüsselung für EMRFS-Daten in Amazon S3 manuell konfigurieren. Weitere Informationen finden Sie unter [Amazon-S3-Verschlüsselung mithilfe von EMRFS-Eigenschaften angeben](#).
- Wenn Sie eine Amazon-EMR-Version vor 5.24.0 verwenden, wird ein verschlüsseltes EBS-Root-Volume nur unterstützt, wenn Sie ein benutzerdefiniertes AMI verwenden. Weitere Informationen finden Sie unter [Erstellen eines benutzerdefinierten AMI mit einem verschlüsselten Amazon EBS-Root-Volume](#) im Verwaltungshandbuch für Amazon EMR.

 Note

Ab Amazon EMR Version 5.24.0 können Sie eine Sicherheitskonfigurationsoption verwenden, um EBS-Root-Geräte und Speichervolumes zu verschlüsseln, wenn Sie dies als Ihren Schlüsselanbieter angeben AWS KMS . Weitere Informationen finden Sie unter [Verschlüsselung lokaler Datenträger](#).

Die Datenverschlüsselung erfordert Aktivierungsschlüssel und Zertifikate. Eine Sicherheitskonfiguration bietet Ihnen die Flexibilität, aus mehreren Optionen zu wählen, darunter Schlüssel AWS Key Management Service, die von Amazon S3 verwaltet werden, sowie Schlüssel und Zertifikate von benutzerdefinierten Anbietern, die Sie bereitstellen. Bei der Nutzung AWS KMS als Schlüsselanbieter fallen Gebühren für die Speicherung und Verwendung von Verschlüsselungsschlüsseln an. Weitere Informationen finden Sie unter [AWS KMS Preise](#).

Bevor Sie die Verschlüsselungsoptionen angeben, legen Sie fest, welche Verwaltungssysteme Sie für die Schlüssel und Zertifikate verwenden möchten. Auf diese Weise können Sie zunächst die Schlüssel und Zertifikate bzw. die von Ihnen bestimmten Anbieter erstellen, die Sie als Teil der Verschlüsselungseinstellungen verwenden möchten.

Verschlüsselung im Ruhezustand von EMRFS-Daten in Amazon S3

Die Amazon S3-Verschlüsselung funktioniert mit Objekten des Amazon EMR File System (EMRFS), die aus Amazon S3 gelesen und in Amazon S3 geschrieben wurden. Sie geben serverseitige Verschlüsselung (SSE) von Amazon S3 oder clientseitige Verschlüsselung (CSE) als Standardverschlüsselungsmodus an, wenn Sie die Verschlüsselung im Ruhezustand aktivieren. Optional können Sie verschiedene Verschlüsselungsmethoden für einzelne Buckets mithilfe von Per bucket encryption overrides (Bucket-weises Überschreiben der Verschlüsselung) angeben. Unabhängig davon, ob Amazon-S3-Verschlüsselung aktiviert ist, verschlüsselt Transport Layer Security (TLS) EMRFS-Objekte bei der Übertragung zwischen EMR-Cluster-Knoten und Amazon S3. Weitere Informationen zur Amazon S3 S3-Verschlüsselung finden Sie unter [Schützen von Daten durch Verschlüsselung](#) im Amazon Simple Storage Service-Benutzerhandbuch.

 Note

Bei der Nutzung AWS KMS fallen Gebühren für die Speicherung und Verwendung von Verschlüsselungsschlüsseln an. Weitere Informationen finden Sie unter [AWS KMS – Preise](#).

Serverseitige Verschlüsselung im Amazon S3

Wenn Sie die Amazon-S3-Verschlüsselung einrichten, verschlüsselt Amazon S3 die Daten auf der Objektebene, während die Daten auf den Datenträger geschrieben werden, und entschlüsselt sie, wenn auf sie zugegriffen wird. Weitere Informationen über SSE finden Sie unter [Schutz von Daten durch serverseitige Verschlüsselung](#) im Amazon Simple Storage Service User Guide.

Wenn Sie SSE in Amazon EMR einrichten, haben Sie die Wahl zwischen zwei verschiedenen Systemen für die Schlüsselverwaltung:

- SSE-S3 – Hierbei verwaltet Amazon S3 die Aktivierungsschlüssel für Sie.
- SSE-KMS — Sie verwenden eine AWS KMS key , um Richtlinien einzurichten, die für Amazon EMR geeignet sind. Weitere Informationen zu den wichtigsten Anforderungen für Amazon EMR finden Sie unter [AWS KMS keys Zur Verschlüsselung verwenden](#).

SSE mit vom Kunden bereitgestellten Schlüsseln (SSE-C) ist für Amazon EMR nicht verfügbar.

Clientseitige Verschlüsselung für Amazon S3

Mit Amazon S3 bei der clientseitigen Verschlüsselung erfolgt der Amazon-S3-Ver- und Entschlüsselungsvorgang im EMRFS-Client auf Ihrem EMR-Cluster. Objekte werden vor dem Hochladen nach Amazon S3 verschlüsselt und nach dem Herunterladen entschlüsselt. Der von Ihnen festgelegte Anbieter stellt den vom Client verwendeten Verschlüsselungsschlüssel bereit. Der Client kann vom AWS KMS bereitgestellte Schlüssel (CSE-KMS) oder eine benutzerdefinierte Java-Klasse verwenden, die den clientseitigen Root-Schlüssel (CSE-C) bereitstellt. Die Verschlüsselungseigenschaften unterscheiden sich geringfügig zwischen CSE-KMS und CSE-C, abhängig vom festgelegten Anbieter und von den Metadaten des Objekts, das entschlüsselt oder verschlüsselt werden soll. Weitere Informationen zu diesen Unterschieden finden Sie unter [Schützen von Daten durch clientseitige Verschlüsselung](#) im Entwicklerhandbuch von Amazon Simple Storage Service.

Note

Amazon S3 CSE stellt nur sicher, dass EMRFS-Daten, die mit Amazon S3 ausgetauscht werden, verschlüsselt sind. Nicht alle Daten auf den Cluster-Instance-Volumes werden verschlüsselt. Da Hue EMRFS nicht verwendet, werden darüber hinaus Objekte, die vom Hue-S3-Dateibrowser in Amazon S3 geschrieben werden, nicht verschlüsselt.

Verschlüsselung im Ruhezustand für Daten in Amazon EMR WAL

Wenn Sie serverseitige Verschlüsselung (SSE) für Write-Ahead-Logging (WAL) einrichten, verschlüsselt Amazon EMR Daten im Ruhezustand. Sie können aus zwei verschiedenen Schlüsselverwaltungssystemen wählen, wenn Sie SSE in Amazon EMR angeben:

SSE-EMR-WAL

Amazon EMR verwaltet Schlüssel für Sie. Standardmäßig verschlüsselt Amazon EMR die Daten, die Sie in Amazon EMR WAL gespeichert haben, mit SSE-EMR-WAL.

SSE-KMS-WAL

Sie verwenden einen AWS KMS Schlüssel, um Richtlinien einzurichten, die für Amazon EMR WAL gelten. Weitere Informationen zur Konfiguration der Verschlüsselung im Ruhezustand für EMR WAL mithilfe eines Kunden-KMS-Schlüssels finden Sie [unter Verschlüsselung im Ruhezustand mithilfe eines Kunden-KMS-Schlüssels für den EMR WAL-Dienst](#).

Note

Sie können Ihren eigenen Schlüssel nicht mit SSE verwenden, wenn Sie WAL mit Amazon EMR aktivieren. Weitere Informationen finden Sie unter [Write-Ahead-Logs \(WAL\) für Amazon EMR](#).

Verschlüsselung lokaler Datenträger

Die folgenden Mechanismen arbeiten zusammen, um lokale Datenträger zu verschlüsseln, wenn Sie die lokale Laufwerksverschlüsselung mithilfe einer Amazon-EMR-Sicherheitskonfiguration aktivieren.

Open-Source-HDFS-Verschlüsselung

HDFS tauscht im Rahmen der verteilten Verarbeitung Daten zwischen Cluster-Instances aus. Außerdem werden Daten aus Instance-Speicher-Volumes und die an Instances angehängten EBS-Volumes gelesen und in sie geschrieben. Wenn Sie die lokale Laufwerksverschlüsselung aktivieren, werden die folgenden Open-Source-Hadoop-Verschlüsselungsoptionen aktiviert:

- [Secure Hadoop RPC](#) ist auf die Option `Privacy` festgelegt, die Simple Authentication Security Layer (SASL) verwendet.

- [Datenverschlüsselung für HDFS-Block-Datenübertragungen](#) ist auf `true` festgelegt und für die Verwendung der AES 256-Verschlüsselung konfiguriert.

Note

Sie können zusätzlich die Apache Hadoop-Verschlüsselung verwenden, indem Sie die Verschlüsselung während der Übertragung aktivieren. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Diese Verschlüsselungseinstellungen aktivieren die transparente HDFS-Verschlüsselung nicht, Sie können sie jedoch manuell konfigurieren. Weitere Informationen finden Sie unter [Transparente Verschlüsselung in HDFS in Amazon EMR](#) in Amazon-EMR-Versionenhinweise.

Instance-Speicher-Verschlüsselung

Bei EC2 Instance-Typen, die NVMe based SSDs als Instance-Speicher-Volume verwenden, wird die NVMe Verschlüsselung unabhängig von den Amazon EMR-Verschlüsselungseinstellungen verwendet. Weitere Informationen finden Sie unter [NVMe SSD-Volumes](#) im EC2 Amazon-Benutzerhandbuch. Für andere Instance-Speicher-Volumes verwendet Amazon EMR unabhängig davon, ob EBS-Volumes mit der EBS-Verschlüsselung oder LUKS verschlüsselt werden, LUKS zum Verschlüsseln des Instance-Speicher-Volumes, wenn die lokale Laufwerksverschlüsselung aktiviert ist.

EBS-Volume-Verschlüsselung

Wenn Sie einen Cluster in einer Region erstellen, in der die EC2 Amazon-Verschlüsselung von EBS-Volumes standardmäßig für Ihr Konto aktiviert ist, werden EBS-Volumes verschlüsselt, auch wenn die lokale Festplattenverschlüsselung nicht aktiviert ist. Weitere Informationen finden Sie unter [Standardverschlüsselung](#) im EC2 Amazon-Benutzerhandbuch. Wenn die lokale Festplattenverschlüsselung in einer Sicherheitskonfiguration aktiviert ist, haben die Amazon EMR-Einstellungen Vorrang vor den EC2 encryption-by-default Amazon-Einstellungen für EC2 Cluster-Instances.

Die folgenden Optionen stehen zum Verschlüsseln von EBS-Volumes mithilfe einer Sicherheitskonfiguration zur Verfügung:

- EBS-Verschlüsselung – Ab Amazon EMR Version 5.24.0 können Sie wählen, ob Sie die EBS-Verschlüsselung aktivieren möchten. Die EBS-Verschlüsselungsoption verschlüsselt das EBS-

Root-Volume und die angefügten Speicher-Volumes. Die EBS-Verschlüsselungsoption ist nur verfügbar, wenn Sie dies AWS Key Management Service als Ihren Schlüsselanbieter angeben. Wir empfehlen die Verwendung der EBS-Verschlüsselung.

- LUKS-Verschlüsselung – Wenn Sie die LUKS-Verschlüsselung für Amazon EBS-Volumes verwenden, gilt die LUKS-Verschlüsselung nur für angeschlossene Speichervolumes, nicht für das Root-Geräte-Volume. Weitere Informationen zur LUKS-Verschlüsselung finden Sie unter [LUKS-Datenträger-Spezifikation](#).

Für Ihren Schlüsselanbieter können Sie eine AWS KMS key mit Richtlinien einrichten, die für Amazon EMR geeignet sind, oder eine benutzerdefinierte Java-Klasse, die die Verschlüsselungsartefakte bereitstellt. Bei der Nutzung AWS KMS fallen Gebühren für die Speicherung und Verwendung von Verschlüsselungsschlüsseln an. Weitere Informationen finden Sie unter [AWS KMS Preise](#).

Note

Um zu überprüfen, ob die EBS-Verschlüsselung auf Ihrem Cluster aktiviert ist, wird empfohlen, den `DescribeVolumes` API-Aufruf zu verwenden. Weitere Informationen finden Sie unter [DescribeVolumes](#). Bei Ausführung von `lsblk` auf dem Cluster wird nur der Status der LUKS-Verschlüsselung anstelle der EBS-Verschlüsselung überprüft.

Verschlüsselung während der Übertragung

Bei der Verschlüsselung während der Übertragung sind mehrere Verschlüsselungsmechanismen aktiviert. Dies sind Open-Source-Funktionen, die anwendungsspezifisch sind und je nach Amazon EMR-Version variieren können. Verwenden Sie in Amazon EMR, um die Verschlüsselung während [the section called “Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI”](#) der Übertragung zu aktivieren. Für EMR-Cluster mit aktivierter Verschlüsselung bei der Übertragung konfiguriert Amazon EMR automatisch die Open-Source-Anwendungskonfigurationen, um die Verschlüsselung bei der Übertragung zu aktivieren. Für erweiterte Anwendungsfälle können Sie Open-Source-Anwendungskonfigurationen direkt konfigurieren, um das Standardverhalten in Amazon EMR zu überschreiben. [Weitere Informationen finden Sie unter Unterstützungsmatrix für Verschlüsselung bei der Übertragung und unter Anwendungen konfigurieren](#).

Im Folgenden finden Sie genauere Informationen zu Open-Source-Anwendungen, die für die Verschlüsselung bei der Übertragung relevant sind:

- Wenn Sie die Verschlüsselung während der Übertragung mit einer Sicherheitskonfiguration aktivieren, aktiviert Amazon EMR die Verschlüsselung während der Übertragung für alle Open-Source-Anwendungsendpunkte, die die Verschlüsselung bei der Übertragung unterstützen. Die Support der Verschlüsselung während der Übertragung für verschiedene Anwendungsendpunkte variiert je nach Amazon EMR-Release-Version. Weitere Informationen finden Sie in der Unterstützungsmatrix für Verschlüsselung bei [der Übertragung](#).
- Sie können Open-Source-Konfigurationen überschreiben, sodass Sie Folgendes tun können:
 - Deaktivieren Sie die Überprüfung des TLS-Hostnamens, wenn Ihre vom Benutzer bereitgestellten TLS-Zertifikate die Anforderungen nicht erfüllen
 - Deaktivieren Sie die Verschlüsselung während der Übertragung für bestimmte Endgeräte je nach Ihren Leistungs- und Kompatibilitätsanforderungen
 - Steuern Sie, welche TLS-Versionen und Cipher Suites verwendet werden sollen.

[Weitere Informationen zu den anwendungsspezifischen Konfigurationen finden Sie in der Unterstützungsmatrix für Verschlüsselung bei der Übertragung](#)

- Neben der Aktivierung der Verschlüsselung bei der Übertragung mit einer Sicherheitskonfiguration erfordern einige Kommunikationskanäle auch zusätzliche Sicherheitskonfigurationen, damit Sie die Verschlüsselung bei der Übertragung aktivieren können. Beispielsweise verwenden einige Open-Source-Anwendungsendpunkte Simple Authentication and Security Layer (SASL) für die Verschlüsselung während der Übertragung, was voraussetzt, dass die Kerberos-Authentifizierung in der Sicherheitskonfiguration des EMR-Clusters aktiviert ist. [Weitere Informationen zu diesen Endpunkten finden Sie in der Unterstützungsmatrix für Verschlüsselung bei der Übertragung](#).
- Wir empfehlen die Verwendung von Software, die TLS v1.2 oder höher unterstützt. Amazon EMR on EC2 liefert die Standard-Corretto JDK-Distribution aus, die festlegt, welche TLS-Versionen, Cipher Suites und Schlüsselgrößen von den Open-Source-Netzwerken, die auf Java laufen, zugelassen werden. Derzeit erzwingen die meisten Open-Source-Frameworks TLS v1.2 oder höher für Amazon EMR 7.0.0 und höhere Versionen. Das liegt daran, dass die meisten Open-Source-Frameworks auf Java 17 für Amazon EMR 7.0.0 und höher laufen. Ältere Amazon EMR-Release-Versionen unterstützen möglicherweise TLS v1.0 und v1.1, da sie ältere Java-Versionen verwenden. Corretto JDK kann jedoch ändern, welche TLS-Versionen von Java unterstützt werden, was sich auf bestehende Amazon EMR-Versionen auswirken kann.

Für die Verwendung der Verschlüsselungsartefakte bei der Verschlüsselung von Daten während der Übertragung stehen Ihnen zwei Optionen zur Verfügung: die Bereitstellung einer ZIP-Datei mit den Zertifikaten, die Sie auf Amazon S3 hochladen, oder der Verweis auf eine benutzerdefinierte Java-Klasse, die Verschlüsselungsartefakte bereitstellt. Weitere Informationen finden Sie unter [Bereitstellen von Zertifikaten für die Verschlüsselung von Daten während der Übertragung mit der Amazon-EMR-Verschlüsselung](#).

Verschlüsselung im Ruhezustand mithilfe eines Kunden-KMS-Schlüssels für den EMR WAL-Service

EMR Write-Ahead Logs (WAL) bieten Kunden Unterstützung für KMS-Schlüssel. encryption-at-rest Im Folgenden wird detailliert beschrieben, wie Amazon EMR WAL integriert AWS KMS ist:

Die EMR-Write-Ahead-Logs (WAL) interagieren AWS bei den folgenden Vorgängen mit: `CreateWAL`, `AppendEdit`, `ArchiveWALCheckPoint`, `CompleteWALFlush`, `DeleteWAL`, `GetCurrentWALTimeReplayEdits`, `EMR_EC2_DefaultRole` standardmäßig `TrimWAL` über das. Wenn eine der zuvor aufgelisteten Operationen aufgerufen wird, führt die EMR-WAL `Decrypt` und `GenerateDataKey` gegen den KMS-Schlüssel aus.

Überlegungen

Beachten Sie Folgendes, wenn Sie die AWS KMS basierte Verschlüsselung für EMR WAL verwenden:

- Die Verschlüsselungskonfiguration kann nicht geändert werden, nachdem eine EMR-WAL erstellt wurde.
- Wenn Sie die KMS-Verschlüsselung mit Ihrem eigenen KMS-Schlüssel verwenden, muss der Schlüssel in derselben Region wie Ihr Amazon EMR-Cluster existieren.
- Sie sind dafür verantwortlich, alle erforderlichen IAM-Berechtigungen aufrechtzuerhalten, und es wird empfohlen, die erforderlichen Berechtigungen während der Laufzeit der WAL nicht zu widerrufen. Andernfalls kann es zu unerwarteten Fehlerszenarien kommen, z. B. wenn EMR WAL nicht gelöscht werden kann, da der zugehörige Verschlüsselungsschlüssel nicht existiert.
- Die Verwendung von AWS KMS Schlüsseln ist mit Kosten verbunden. Weitere Informationen finden Sie unter [AWS Key Management Service Preise](#).

Erforderliche IAM-Berechtigungen

Um Ihren Kunden-KMS-Schlüssel zum Verschlüsseln von EMR WAL im Ruhezustand zu verwenden, müssen Sie sicherstellen, dass Sie die richtigen Berechtigungen für die EMR WAL-Clientrolle und den EMR WAL-Dienstprinzipal festlegen. `emrwal.amazonaws.com`

Berechtigungen für die EMR WAL-Clientrolle

Im Folgenden finden Sie die IAM-Richtlinie, die für die EMR-WAL-Clientrolle erforderlich ist:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey*",
    ],
    "Resource": "*"
  }
}
```

Der EMR-WAL-Client auf dem EMR-Cluster wird `EMR_EC2_DefaultRole` standardmäßig verwendet. Wenn Sie eine andere Rolle für das Instanzprofil im EMR-Cluster verwenden, stellen Sie sicher, dass jede Rolle über die entsprechenden Berechtigungen verfügt.

Weitere Informationen zur Verwaltung der Rollenrichtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Berechtigungen für die KMS-Schlüsselrichtlinie

In Ihrer KMS-Richtlinie müssen Sie der EMR-WAL-Clientrolle und dem EMR-WAL-Dienst `Decrypt` und die entsprechenden `GenerateDataKey*` Berechtigungen zuweisen. Weitere Informationen zur Verwaltung von Schlüsselrichtlinien finden Sie unter [KMS-Schlüsselrichtlinie](#).

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

```

    "AWS": [
      "arn:aws:iam::accountID:role/EMR_EC2_DefaultRole"
    ],
    "Service": [
      "emrwal.amazonaws.com"
    ]
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": [
    "*"
  ]
}
]
}

```

Die im Snippet angegebene Rolle kann sich ändern, wenn Sie die Standardrolle ändern.

Überwachung der Amazon EMR WAL-Interaktion mit AWS KMS

Amazon EMR WAL-Verschlüsselungskontext

Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, der beliebige, nicht geheime Daten enthält. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten einbeziehen, wird der Verschlüsselungskontext AWS KMS kryptografisch an die verschlüsselten Daten gebunden. Zur Entschlüsselung der Daten müssen Sie denselben Verschlüsselungskontext übergeben.

In seinen [GenerateDataKey](#) und [Decrypt-Anfragen](#) an AWS KMS verwendet Amazon EMR WAL einen Verschlüsselungskontext mit einem Name-Wert-Paar, das den EMR-WAL-Namen identifiziert.

```

"encryptionContext": {
  "aws:emrwal:walname": "111222333444555-testworkspace-emrwalclustertest-
emrwaltestwalname"
}

```

Sie können den Verschlüsselungskontext verwenden, um diese kryptografischen Vorgänge in Prüfaufzeichnungen und Protokollen wie [Amazon CloudWatch Logs](#) zu identifizieren AWS CloudTrail und als Voraussetzung für die Autorisierung in Richtlinien und Zuschüssen zu verwenden.

Schlüssel und Zertifikate für die Datenverschlüsselung mit Amazon EMR erstellen

Bevor Sie Verschlüsselungsoptionen unter Verwendung einer Sicherheitskonfiguration angeben, legen Sie zunächst den Anbieter der Schlüssel und Verschlüsselungsartefakte fest. Sie können beispielsweise einen benutzerdefinierten Anbieter verwenden AWS KMS , den Sie erstellen. Erstellen Sie als Nächstes die erforderlichen Schlüssel oder den Schlüsselanbieter, wie in diesem Abschnitt beschrieben.

Bereitstellung von Schlüsseln für die Verschlüsselung von Daten im Ruhezustand

Sie können AWS Key Management Service (AWS KMS) oder einen benutzerdefinierten Schlüsselanbieter für die Verschlüsselung von Daten im Ruhezustand in Amazon EMR verwenden. Bei der Nutzung AWS KMS fallen Gebühren für die Speicherung und Verwendung von Verschlüsselungsschlüsseln an. Weitere Informationen finden Sie unter [AWS KMS Preise](#).

In diesem Thema finden Sie Schlüsselrichtliniendetails für KMS Schlüssel zur Verwendung mit Amazon EMR sowie Anleitungen und Codebeispiele für das Schreiben einer benutzerdefinierten Schlüsselanbieterklasse für die Amazon S3-Verschlüsselung. Weitere Informationen zum Erstellen von -Schlüsseln finden Sie unter [Erstellen von Schlüsseln](#) im AWS Key Management Service - Entwicklerhandbuch.

AWS KMS keys Für die Verschlüsselung verwenden

Der AWS KMS Verschlüsselungsschlüssel muss in derselben Region wie Ihre Amazon EMR-Cluster-Instance und die mit EMRFS verwendeten Amazon S3 S3-Buckets erstellt werden. Wenn sich der von Ihnen angegebene Schlüssel in einem anderen Konto befindet als dem, das Sie zur Konfiguration eines Clusters verwenden, müssen Sie den Schlüssel mit seinem ARN angeben.

Die Rolle für das EC2 Amazon-Instance-Profil muss über Berechtigungen zur Verwendung des von Ihnen angegebenen KMS-Schlüssels verfügen. Die Standardrolle für das Instance-Profil in Amazon EMR ist `EMR_EC2_DefaultRole`. Wenn Sie eine andere Rolle für das Instance-Profil oder IAM-Rollen für EMRFS-Anfragen an Amazon S3 verwenden, stellen Sie sicher, dass jede Rolle je nach Bedarf als Schlüsselbenutzer hinzugefügt wird. So erhält die Rolle die Berechtigung, den KMS-Schlüssel zu verwenden. Weitere Informationen finden Sie unter [Nutzung von Schlüsselrichtlinien](#) im AWS Key Management Service -Entwicklerhandbuch und [Konfigurieren von IAM-Rollen für EMRFS-Anfragen an Amazon S3](#).

Sie können das verwenden AWS Management Console , um Ihr Instance-Profil oder EC2 Instance-Profil zur Liste der Hauptbenutzer für den angegebenen KMS-Schlüssel hinzuzufügen, oder Sie

können das AWS CLI oder ein AWS SDK verwenden, um eine entsprechende Schlüsselrichtlinie anzuhängen.

Hinweis: Amazon EMR unterstützt nur [symmetrische KMS-Schlüssel](#). Sie können keinen [asymmetrischen KMS-Schlüssel](#) verwenden, um Data-at-Rest in einem Amazon-EMR-Cluster zu verschlüsseln. Wie Sie feststellen, ob ein KMS-Schlüssel symmetrisch oder asymmetrisch ist, erfahren Sie unter [Erkennen symmetrischer und asymmetrischer KMS-Schlüssel](#).

Im folgenden Verfahren wird beschrieben, wie Sie das Amazon-EMR-Instance-Profil mithilfe der `EMR_EC2_DefaultRole` als Schlüsselbenutzer mit AWS Management Console hinzufügen. Dabei wird davon ausgegangen, dass Sie bereits einen KMS-Schlüssel erstellt haben. Weitere Informationen über die Erstellung eines neuen KMS-Schlüssels finden Sie unter [Erstellen von Schlüsseln](#) im AWS Key Management Service -Entwicklerhandbuch.

Um das EC2 Instance-Profil für Amazon EMR zur Liste der Benutzer von Verschlüsselungsschlüsseln hinzuzufügen

1. Melden Sie sich bei der AWS Key Management Service (AWS KMS) -Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/kms>.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie den Alias des zu ändernden KMS-Schlüssels aus.
4. Wählen Sie auf der Seite mit den Schlüsseldetails unter Key Users (Schlüsselbenutzer(die Option Add (Hinzufügen) aus.
5. Wählen Sie die entsprechende Rolle im Dialogfeld Add key users (Schlüsselbenutzer hinzufügen) aus. Der Name der Standardrolle lautet `EMR_EC2_DefaultRole`.
6. Wählen Sie Hinzufügen aus.

Aktivieren der EBS-Verschlüsselung durch Bereitstellung zusätzlicher Berechtigungen für KMS-Schlüssel

Ab Amazon EMR Version 5.24.0 können Sie EBS-Root-Volumes und Speicher-Volumes mithilfe einer Sicherheitskonfigurationsoption verschlüsseln. Um diese Option zu aktivieren, müssen Sie AWS KMS als Ihren Schlüsselanbieter angeben. Darüber hinaus müssen Sie der Servicerolle die von `EMR_DefaultRole` Ihnen angegebenen Berechtigungen zur Verwendung der von AWS KMS key Ihnen angegebenen Rechte erteilen.

Sie können das verwenden AWS Management Console , um die Servicerolle zur Liste der Hauptbenutzer für den angegebenen KMS-Schlüssel hinzuzufügen, oder Sie können das AWS CLI oder ein AWS SDK verwenden, um eine entsprechende Schlüsselrichtlinie anzuhängen.

Das folgende Verfahren beschreibt, wie Sie die AWS Management Console standardmäßige Amazon EMR-Servicerolle `EMR_DefaultRole` als Schlüsselbenutzer hinzufügen können. Dabei wird davon ausgegangen, dass Sie bereits einen KMS-Schlüssel erstellt haben. Weitere Informationen über die Erstellung eines neuen KMS Schlüssels finden Sie unter [Erstellen von Schlüsseln](#) im AWS Key Management Service Entwicklerhandbuch.

So fügen Sie die Amazon EMR-Servicerolle zur Liste der Benutzer von Verschlüsselungsschlüsseln hinzu

1. Melden Sie sich bei der AWS Key Management Service (AWS KMS) -Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/kms>.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Wählen Sie links Vom Kunden verwaltete Schlüssel aus.
4. Wählen Sie den Alias des zu ändernden KMS-Schlüssels aus.
5. Wählen Sie auf der Seite mit den Schlüsseldetails unter Key Users (Schlüsselbenutzer(die Option Add (Hinzufügen) aus.
6. Wählen Sie im Abschnitt „Hauptbenutzer hinzufügen“ die entsprechende Rolle aus. Der Name der Standard-Servicerolle für Amazon EMR lautet `EMR_DefaultRole`.
7. Wählen Sie Hinzufügen aus.

Erstellen eines benutzerdefinierten Schlüsselanbieters

Wenn Sie eine Sicherheitskonfiguration verwenden, müssen Sie einen anderen Anbieterklassennamen für die Verschlüsselung lokaler Datenträger und die Amazon-S3-Verschlüsselung angeben. Die Anforderungen für den benutzerdefinierten Schlüsselanbieter hängen davon ab, ob Sie die lokale Festplattenverschlüsselung und die Amazon S3 S3-Verschlüsselung sowie die Amazon EMR-Release-Version verwenden.

Abhängig von der Art der Verschlüsselung, die Sie bei der Erstellung eines benutzerdefinierten Schlüsselanbieters verwenden, muss die Anwendung auch unterschiedliche `EncryptionMaterialsProvider` Schnittstellen implementieren. Beide Schnittstellen sind im AWS SDK for Java Version 1.11.0 und höher verfügbar.

- Um die Amazon S3 S3-Verschlüsselung zu implementieren, verwenden Sie das Modell [com.amazonaws.services.s3.model. EncryptionMaterialsProvider Schnittstelle](#).
- Verwenden Sie die Datei [com.amazonaws.services.elasticmapreduce.spi.security](#), um die lokale Festplattenverschlüsselung zu implementieren. [EncryptionMaterialsProvider Schnittstelle](#).

Sie können jede Strategie verwenden, um Verschlüsselungsmaterial für die Implementierung bereitzustellen. Sie können sich beispielsweise dafür entscheiden, statisches Verschlüsselungsmaterial bereitzustellen oder es in ein komplexeres Schlüsselverwaltungssystem zu integrieren.

Wenn Sie die Amazon S3 S3-Verschlüsselung verwenden, müssen Sie die Verschlüsselungsalgorithmen AES/GCM/NoPadding für benutzerdefinierte Verschlüsselungsmaterialien verwenden.

Wenn Sie die lokale Festplattenverschlüsselung verwenden, variiert der Verschlüsselungsalgorithmus, der für benutzerdefinierte Verschlüsselungsmaterialien verwendet werden soll, je nach EMR-Version. Für Amazon EMR 7.0.0 und niedriger müssen Sie verwenden. AES/GCM/NoPadding Für Amazon EMR 7.1.0 und höher müssen Sie AES verwenden.

Die EncryptionMaterialsProvider Klasse ruft Verschlüsselungsmaterial anhand des Verschlüsselungskontextes ab. Amazon EMR aktualisiert die Verschlüsselungskontextinformationen während der Laufzeit. So wird der Aufrufer bei der Entscheidung unterstützt, welche korrekten Verschlüsselungsmaterialien zurückzugeben sind.

Example Beispiel: Verwenden eines benutzerdefinierten Schlüsselanbieters für die Amazon-S3-Verschlüsselung mit EMRFS

Wenn Amazon EMR die Verschlüsselungsmaterialien von der EncryptionMaterialsProvider Klasse abrufen, um die Verschlüsselung durchzuführen, füllt EMRFS optional das Argument MaterialsDescription mit zwei Feldern auf: dem Amazon S3 S3-URI für das Objekt und dem des Clusters, die JobFlowId von der Klasse verwendet werden können, um Verschlüsselungsmaterialien selektiv zurückzugeben. EncryptionMaterialsProvider

Beispielsweise kann der Anbieter unterschiedliche Schlüssel für unterschiedliche Amazon-S3-URI-Präfixe zurückgeben. Es ist die Beschreibung der zurückgegebenen Verschlüsselungsmaterialien, die schließlich mit dem Amazon-S3-Objekt gespeichert wird, und nicht der materialsDescription-Wert, der von EMRFS generiert und an den Anbieter weitergeleitet wird. Beim Entschlüsseln eines Amazon S3 S3-Objekts wird die Beschreibung des Verschlüsselungsmaterials an die EncryptionMaterialsProvider

Klasse übergeben, sodass sie wiederum selektiv den passenden Schlüssel zur Entschlüsselung des Objekts zurückgeben kann.

Eine `EncryptionMaterialsProvider` Referenzimplementierung finden Sie weiter unten. Ein weiterer benutzerdefinierter Anbieter [EMRFSRSAEncryptionMaterialsProvider](#), ist erhältlich bei GitHub.

```
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.EncryptionMaterialsProvider;
import com.amazonaws.services.s3.model.KMSEncryptionMaterials;
import org.apache.hadoop.conf.Configurable;
import org.apache.hadoop.conf.Configuration;

import java.util.Map;

/**
 * Provides KMSEncryptionMaterials according to Configuration
 */
public class MyEncryptionMaterialsProviders implements EncryptionMaterialsProvider,
    Configurable{
    private Configuration conf;
    private String kmsKeyId;
    private EncryptionMaterials encryptionMaterials;

    private void init() {
        this.kmsKeyId = conf.get("my.kms.key.id");
        this.encryptionMaterials = new KMSEncryptionMaterials(kmsKeyId);
    }

    @Override
    public void setConf(Configuration conf) {
        this.conf = conf;
        init();
    }

    @Override
    public Configuration getConf() {
        return this.conf;
    }

    @Override
    public void refresh() {

    }
```

```
@Override
public EncryptionMaterials getEncryptionMaterials(Map<String, String>
materialsDescription) {
    return this.encryptionMaterials;
}

@Override
public EncryptionMaterials getEncryptionMaterials() {
    return this.encryptionMaterials;
}
}
```

Bereitstellen von Zertifikaten für die Verschlüsselung von Daten während der Übertragung mit der Amazon-EMR-Verschlüsselung

Mit Version Amazon EMR 4.8.0 oder höher haben Sie zwei Möglichkeiten für die Angabe von Artefakten für die Verschlüsselung von Daten während der Übertragung mithilfe einer Sicherheitskonfiguration:

- Sie können PEM-Zertifikate manuell erstellen, diese in einer ZIP-Datei einschließen und anschließend in Amazon S3 auf die ZIP-Datei verweisen.
- Sie können einen benutzerdefinierten Zertifikatanbieter als Java-Klasse implementieren. Geben Sie dazu die JAR-Datei der Anwendung in Amazon S3 an und nennen Sie anschließend den vollständigen Klassennamen des Anbieters, wie in der Anwendung deklariert. Die Klasse muss die [TLSArtifactsProvider-Schnittstelle](#) implementieren, die ab AWS SDK für Java Version 1.11.0 verfügbar ist.

Amazon EMR lädt automatisch Artefakte auf jeden Knoten im Cluster herunter und verwendet sie später dazu, um die Open-Source-Features für die Verschlüsselung von Daten während der Übertragung zu implementieren. Weitere Informationen zu den verfügbaren Optionen finden Sie unter [Verschlüsselung während der Übertragung](#).

Verwenden der PEM-Zertifikate

Wenn Sie eine ZIP-Datei für die Verschlüsselung von Daten während der Übertragung angeben, müssen die PEM-Dateien innerhalb der ZIP-Datei für die Sicherheitskonfiguration genau wie nachfolgend angegeben benannt sein:

Zertifikate für Verschlüsselung von Daten während der Übertragung

Dateiname	Erforderlich/optional	Details
privateKey.pem	Erforderlich	Privater Schlüssel
certificateChain.pem	Erforderlich	Zertifikatskette
trustedCertificate.s.pem	Optional	Es wird empfohlen, ein Zertifikat bereitzustellen, das nicht von der standardmäßig vertrauenswürdigen Stammzertifizierungsstelle (CA) von Java signiert ist, oder von einer Zwischenzertifizierungsstelle, die eine Verbindung zur standardmäßig vertrauenswürdigen Stammzertifizierungsstelle von Java herstellen kann. Wir empfehlen nicht, public zu verwenden, wenn Sie Platzhalterzertifikate verwenden oder CAs wenn Sie die Hostnamenüberprüfung deaktivieren.

Möglicherweise sollten Sie die private Schlüssel-PEM-Datei als Platzhalterzertifikat konfigurieren, so gewähren Sie Zugriff auf die Amazon VPC-Domain, in der Ihre Cluster-Instances gespeichert sind. Wenn sich Ihr Cluster beispielsweise in der Region us-east-1 (N. Virginia) befindet, könnten Sie in der Zertifikatskonfiguration einen allgemeinen Namen angeben, der durch die Angabe von `CN=*.ec2.internal` in der Zertifikatssubjektdefinition Zugriff auf den Cluster gewährt. Wenn sich Ihr Cluster in der Region us-west-2 (Oregon) befindet, könnten Sie `CN=*.us-west-2.compute.internal` angeben.

Wenn die bereitgestellte PEM-Datei im Verschlüsselungsartefakt kein Platzhalterzeichen im allgemeinen Namen für die Domäne enthält, müssen Sie den Wert von `to` ändern.

`hadoop.ssl.hostname.verifier` `ALLOW_ALL` Fügen Sie dazu in den Amazon EMR-Versionen 7.3.0 und höher die `core-site` Klassifizierung hinzu, wenn Sie Konfigurationen an einen Cluster senden. Fügen Sie in Versionen vor 7.3.0 die Konfiguration `"hadoop.ssl.hostname.verifier": "ALLOW_ALL"` direkt zur Datei hinzu. `core-site.xml` Diese Änderung ist erforderlich, da die standardmäßige Hostnamen-Verifizierung einen Hostnamen ohne Platzhalter erfordert, da ihn alle Hosts im Cluster verwenden. Weitere Informationen zur EMR-Clusterkonfiguration innerhalb einer Amazon VPC finden Sie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#)

Das folgende Beispiel zeigt, wie [OpenSSL](#) verwendet wird, um ein selbstsigniertes X.509-Zertifikat mit einem privaten 2048-Bit-RSA-Schlüssel zu generieren. Der Schlüssel ermöglicht den Zugriff auf die Amazon-EMR-Cluster-Instances des Ausstellers in der Region `us-west-2` (Oregon), wie durch den Domainnamen `*.us-west-2.compute.internal` als allgemeiner Name angegeben.

Es können weitere optionale Subjektelemente wie Land (Country, C), Status (Status, S), Gebietsschema (Locale, L) usw. angegeben werden. Da ein selbstsigniertes Zertifikat generiert wird, kopiert der zweite Befehl im Beispiel die Datei `certificateChain.pem` zur Datei `trustedCertificates.pem`. Der dritte Befehl verwendet `zip` zum Erstellen der Datei `my-certs.zip`, die die Zertifikate enthält.

Important

Dieses Beispiel dient nur zur Veranschaulichung. proof-of-concept Die Verwendung von selbstsignierten Zertifikaten wird nicht empfohlen und stellt ein potenzielles Sicherheitsrisiko dar. Verwenden Sie eine vertrauenswürdige Zertifizierungsstelle (CA), um die Zertifikate für Produktionssysteme auszustellen.

```
$ openssl req -x509 -newkey rsa:2048 -keyout privateKey.pem -out certificateChain.pem  
-days 365 -nodes -subj '/C=US/ST=Washington/L=Seattle/O=MyOrg/OU=MyDept/CN=*.us-  
west-2.compute.internal'  
$ cp certificateChain.pem trustedCertificates.pem  
$ zip -r -X my-certs.zip certificateChain.pem privateKey.pem trustedCertificates.pem
```

Grundlegendes zur Verschlüsselung bei der Übertragung

Sie können einen EMR-Cluster so konfigurieren, dass er Open-Source-Frameworks wie [Apache Spark](#), [Apache Hive](#) und [Presto](#) ausführt. Jedes dieser Open-Source-Frameworks hat eine Reihe

von Prozessen, die auf den EC2 Instanzen eines Clusters ausgeführt werden. Jeder dieser Prozesse kann Netzwerkendpunkte für die Netzwerkkommunikation hosten.

Wenn die Verschlüsselung während der Übertragung auf einem EMR-Cluster aktiviert ist, verwenden verschiedene Netzwerkendpunkte unterschiedliche Verschlüsselungsmechanismen. In den folgenden Abschnitten erfahren Sie mehr über die spezifischen Open-Source-Framework-Netzwerkendpunkte, die mit der Verschlüsselung während der Übertragung unterstützt werden, die zugehörigen Verschlüsselungsmechanismen und welche Amazon EMR-Version diese Unterstützung hinzugefügt hat. Jede Open-Source-Anwendung verfügt möglicherweise auch über unterschiedliche Best Practices und Open-Source-Framework-Konfigurationen, die Sie ändern können.

Um die Verschlüsselung während der Übertragung bestmöglich abzudecken, empfehlen wir, sowohl die Verschlüsselung während der Übertragung als auch Kerberos zu aktivieren. Wenn Sie nur die Verschlüsselung bei der Übertragung aktivieren, ist die Verschlüsselung bei der Übertragung nur für die Netzwerkendpunkte verfügbar, die TLS unterstützen. Kerberos ist erforderlich, da einige Open-Source-Framework-Netzwerkendpunkte Simple Authentication and Security Layer (SASL) für die Verschlüsselung während der Übertragung verwenden.

Beachten Sie, dass Open-Source-Frameworks, die in Amazon EMR 7.x.x-Versionen nicht unterstützt werden, nicht enthalten sind.

Spark

Wenn Sie die Verschlüsselung während der Übertragung in Sicherheitskonfigurationen aktivieren, `spark.authenticate` wird automatisch die AES-basierte Verschlüsselung für `true` RPC-Verbindungen ausgewählt und verwendet.

Ab Amazon EMR 7.3.0 können Sie Spark-Anwendungen, die vom Hive-Metastore abhängen, nicht mehr verwenden, wenn Sie Verschlüsselung bei der Übertragung und Kerberos-Authentifizierung verwenden. [Hive 3 behebt dieses Problem in HIVE-16340](#). [HIVE-44114](#) behebt dieses Problem vollständig, wenn Open-Source-Spark auf Hive 3 aktualisiert werden kann. In der Zwischenzeit können Sie sich daran machen, dieses Problem `hive.metastore.use.SSL` zu `false` umgehen. Weitere Informationen finden Sie unter [Konfigurieren von Anwendungen](#).

Weitere Informationen finden Sie unter [Spark-Sicherheit](#) in der Apache Spark-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Spark History Server	spark.ssl.history.port	18480	TLS	emr-5.3.0+, emr-6.0.0+, emr-7.0.0+
Spark-Benutzeroberfläche	spark.ui.port	4440	TLS	emr-5.3.0+, emr-6.0.0+, emr-7.0.0+
Spark-Treiber	spark.driver.port	Dynamisch	AES-basierte Verschlüsselung auf Spark-Basis	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
Spark Executor	Executor-Port (keine benannte Konfiguration)	Dynamisch	AES-Basierte Verschlüsselung auf Spark-Basis	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
GARN NodeManager	spark.shuffle.service.port ¹	7337	AES-basierte Verschlüsselung auf Spark-Basis	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+

¹ wird auf YARN `spark.shuffle.service.port` gehostet, aber nur von Apache NodeManager Spark verwendet.

Hadoop-GARN

[Secure Hadoop RPC ist auf](#) SASL-basierte Verschlüsselung bei der Übertragung eingestellt `privacy` und verwendet diese. Dies setzt voraus, dass die Kerberos-Authentifizierung in der Sicherheitskonfiguration aktiviert ist. Wenn Sie keine Verschlüsselung während der Übertragung für Hadoop RPC wünschen, konfigurieren Sie `hadoop.rpc.protection = authentication`. Wir empfehlen, die Standardkonfiguration für maximale Sicherheit zu verwenden.

Wenn Ihre TLS-Zertifikate die Anforderungen zur Überprüfung des TLS-Hostnamens nicht erfüllen, können Sie sie konfigurieren `hadoop.ssl.hostname.verifier = ALLOW_ALL`. Wir empfehlen Ihnen, die Standardkonfiguration von `hadoop.ssl.hostname.verifier = DEFAULT` zu verwenden, die die Überprüfung des TLS-Hostnamens erzwingt.

Um HTTPS für die Endpunkte der YARN-Webanwendung zu deaktivieren, konfigurieren Sie `yarn.http.policy = HTTP_ONLY`. Dadurch bleibt der Datenverkehr zu diesen Endpunkten unverschlüsselt. Wir empfehlen, die Standardkonfiguration für maximale Sicherheit zu verwenden.

Weitere Informationen finden Sie unter [Hadoop in Secure Mode](#) in der Apache-Hadoop-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
ResourceManager	<code>yarn.resourcemanager.webapp.address</code>	8088	TLS	emr-7.3.0+
ResourceManager	<code>yarn.resourcemanager.resource-tracker.address</code>	8025	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
ResourceManager	<code>yarn.resourcemanager</code>	8030	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+,

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
	er.scheduler.address			emr-6.0.0+, emr-7.0.0+
ResourceManager	yarn.resourcemanager.address	8032	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
ResourceManager	yarn.resourcemanager.admin.address	8033	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
TimelineServer	garn.timeline-service.adresse	10200	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
TimelineServer	garn.timeline-service.webapp.adresse	8188	TLS	emr-7.3.0+
WebApplicationProxy	yarn.web-proxy.address	20888	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
NodeManager	yarn.node.manager.address	8041	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
NodeManager	yarn.node.manager.localizer.address	8040	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
NodeManager	yarn.node.manager.webapp.address	8044	TLS	emr-7.3.0+
NodeManager	mapreduce.shuffle. Anschluss 1	13562	TLS	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
NodeManager	spark.shuffle.service.port ²	7337	AES-basierte Verschlüsselung auf Spark-Basis	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+

¹ wird auf `mapreduce.shuffle.port` YARN gehostet, aber nur von Hadoop NodeManager verwendet.

MapReduce

² `spark.shuffle.service.port` wird auf YARN gehostet NodeManager, aber nur von Apache Spark verwendet.

Bekanntes Problem

Die `yarn.log.server.url` Konfiguration in verwendet derzeit HTTP mit Port 19888, wodurch der Zugriff auf Anwendungsprotokolle über die Resource Manager-Benutzeroberfläche verhindert wird.
Betrifft Version: EMR - 7.3.0 bis EMR - 7.8.0.

Verwenden Sie die folgende Problemumgehung:

1. Ändern Sie die `yarn.log.server.url` Konfiguration `soyarn-site.xml`, dass das HTTPS Protokoll und die Portnummer 19890 verwendet werden.
2. Starten Sie YARN Resource Manager neu:
`sudo systemctl restart hadoop-yarn-resourcemanager.service.`

Hadoop HDFS

Der Hadoop-Name-Node, der Datenknoten und der Journal-Node unterstützen standardmäßig TLS, wenn die Verschlüsselung während der Übertragung in EMR-Clustern aktiviert ist.

[Secure Hadoop RPC ist auf „auf“](#) eingestellt `privacy` und verwendet eine SASL-basierte Verschlüsselung bei der Übertragung. Dies setzt voraus, dass die Kerberos-Authentifizierung in der Sicherheitskonfiguration aktiviert ist.

Wir empfehlen, die für HTTPS-Endpunkte verwendeten Standardports nicht zu ändern.

[Die Datenverschlüsselung bei der HDFS-Blockübertragung verwendet](#) AES 256 und erfordert, dass die Verschlüsselung im Ruhezustand in der Sicherheitskonfiguration aktiviert ist.

Weitere Informationen finden Sie unter [Hadoop in Secure Mode](#) in der Apache-Hadoop-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Namenode	<code>dfs.namenode.https-Adresse</code>	9871	TLS	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Benennen Sie den Knoten	dfs.namenode.rpc-address	8020	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
Datenknoten	dfs.datanode.https.address	9865	TLS	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
Datenknoten	dfs.datanode.address	9866	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
Journalknoten	dfs.journalnode.https-Adresse	8481	TLS	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+
Journalknoten	dfs.journalnode.rpc-Adresse	8485	SASL + Kerberos	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
DFSZKFailoverSteuerung	dfs.ha.zkfc.port	8019	Keine	TLS für ZKFC wird nur in Hadoop 3.4.0 unterstützt. Weitere Informationen finden Sie unter HADOOP-18919 . Amazon EMR Version 7.1.0 befindet sich derzeit auf Hadoop 3.3.6. Höhere Amazon EMR-Versionen sind in future auf Hadoop 3.4.0 verfügbar

Hadoop MapReduce

Hadoop MapReduce, Job History Server und MapReduce Shuffle unterstützen standardmäßig TLS, wenn die Verschlüsselung während der Übertragung in EMR-Clustern aktiviert ist.

[Hadoop-verschlüsselter Shuffle verwendet MapReduce](#) TLS.

Wir empfehlen, die Standardports für HTTPS-Endpunkte nicht zu ändern.

Weitere Informationen finden Sie unter [Hadoop in Secure Mode](#) in der Apache-Hadoop-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
JobHistoryServer	mapreduce.jobhistory.webapp.https.address	19890	TLS	emr-7.3.0+
GARN NodeManager	mapreduce.shuffle.port 1	13562	TLS	emr-4.8.0+, emr-5.0.0+, emr-6.0.0+, emr-7.0.0+

1 wird auf `mapreduce.shuffle.port` YARN gehostet, aber nur von Hadoop NodeManager verwendet.

MapReduce

Presto

In Amazon EMR-Versionen 5.6.0 und höher verwendet die interne Kommunikation zwischen dem Presto-Koordinator und den Mitarbeitern TLS. Amazon EMR richtet alle erforderlichen Konfigurationen ein, um eine [sichere interne](#) Kommunikation in Presto zu ermöglichen.

Wenn der Connector den Hive-Metastore als Metadatenpeicher verwendet, wird die Kommunikation zwischen dem Communicator und dem Hive-Metastore ebenfalls mit TLS verschlüsselt.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Presto-Koordinator	http-server.https.port	8446	TLS	emr-5.6.0+, emr-6.0.0+, emr-7.0.0+

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Presto Arbeiter	http-server.https.port	8446	TLS	emr-5.6.0+, emr-6.0.0+, emr-7.0.0+

Trino

In Amazon EMR-Versionen 6.1.0 und höher verwendet die interne Kommunikation zwischen dem Presto-Koordinator und den Mitarbeitern TLS. Amazon EMR richtet alle erforderlichen Konfigurationen ein, um eine [sichere interne](#) Kommunikation in Trino zu ermöglichen.

Wenn der Connector den Hive-Metastore als Metadatenpeicher verwendet, wird die Kommunikation zwischen dem Communicator und dem Hive-Metastore ebenfalls mit TLS verschlüsselt.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Trino-Koordinator	http-server.https.port	8446	TLS	emr-6.1.0+, emr-7.0.0+
Trino Arbeiter	http-server.https.port	8446	TLS	emr-6.1.0+, emr-7.0.0+

Hive und Tez

Standardmäßig unterstützen Hive Server 2, Hive Metastore Server, Hive LLAP Daemon Web UI und Hive LLAP shuffle alle TLS, wenn die Verschlüsselung während der Übertragung in den EMR-Clustern aktiviert ist. [Weitere Informationen zu den Hive-Konfigurationen finden Sie unter Konfigurationseigenschaften.](#)

Die Tez-Benutzeroberfläche, die auf dem Tomcat-Server gehostet wird, ist ebenfalls HTTPS-fähig, wenn die Verschlüsselung während der Übertragung im EMR-Cluster aktiviert ist. HTTPS ist jedoch für den Tez AM-Web-UI-Dienst deaktiviert, sodass AM-Benutzer keinen Zugriff auf die Keystore-Datei für den öffnenden SSL-Listener haben. Sie können dieses Verhalten auch mit den booleschen Konfigurationen und aktivieren. `tez.am.tez-ui.webservice.enable.ssl` `tez.am.tez-ui.webservice.enable.client.auth`

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
HiveServer2	hive.server2.thrift.port	10000	TLS	emr-6.9.0+, emr-7.0.0+
HiveServer2	hive.server2.thrift.http.port	10001	TLS	emr-6.9.0+, emr-7.0.0+
HiveServer2	hive.server2.webui.port	10002	TLS	emr-7.3.0+
HiveMetastoreServer	hive.metastore.port	9083	TLS	emr-7.3.0+
LLAP-Daemon	hive.llap.daemon.yarn.shuffle.port	1551	TLS	emr-7.3.0+
LLAP-Daemon	hive.llap.daemon.web.port	15002	TLS	emr-7.3.0+
LLAP-Daemon		15003	Keine	Hive unterstützt für diesen

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
	hive.llap.daemon.output.service.port			Endpunkt keine Verschlüsselung bei der Übertragung
LLAP-Daemon	hive.llap.management.rpc.port	15004	Keine	Hive unterstützt für diesen Endpunkt keine Verschlüsselung bei der Übertragung
LLAP-Daemon	hive.llap.plugin.rpc.port	Dynamisch	Keine	Hive unterstützt für diesen Endpunkt keine Verschlüsselung bei der Übertragung
LLAP-Daemon	hive.llap.daemon.rpc.port	Dynamisch	Keine	Hive unterstützt für diesen Endpunkt keine Verschlüsselung bei der Übertragung
Internet HCat	templeton.port	50111	TLS	emr-7.3.0+

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Tez-Anwendungsmaster	tez.am.client.am.port-Bereich tez.am.task.am.port-Bereich	Dynamisch	Keine	Tez unterstützt für diesen Endpunkt keine Verschlüsselung bei der Übertragung
Tez Application Master	tez.am.tez-ui.webservice.portrange	Dynamisch	Keine	Standardmäßig deaktiviert. Kann mithilfe von Tez-Konfigurationen in emr-7.3.0+ aktiviert werden
Tez-Aufgabe	N/A - nicht konfigurierbar	Dynamisch	Keine	Tez unterstützt für diesen Endpunkt keine Verschlüsselung bei der Übertragung
Tez UI	Konfigurierbar über den Tomcat-Server, auf dem die Tez-Benutzeroberfläche gehostet wird	8080	TLS	emr-7.3.0+

Flink

Apache Flink REST-Endpunkte und die interne Kommunikation zwischen Flink-Prozessen unterstützen TLS standardmäßig, wenn Sie die Verschlüsselung während der Übertragung in EMR-Clustern aktivieren.

[security.ssl.internal.enabled](#) ist auf In-Transit-Verschlüsselung eingestellt `true` und verwendet diese für die interne Kommunikation zwischen den Flink-Prozessen. Wenn Sie keine Verschlüsselung während der Übertragung für die interne Kommunikation wünschen, deaktivieren Sie diese Konfiguration. Wir empfehlen Ihnen, die Standardkonfiguration für maximale Sicherheit zu verwenden.

Amazon EMR stellt [security.ssl.rest.enabled](#) die Verschlüsselung während der Übertragung für die REST-Endpunkte ein `true` und verwendet diese. Darüber hinaus setzt Amazon EMR auch [historyserver.web.ssl.enabled](#) auf `true`, um die TLS-Kommunikation mit dem Flink-Verlaufsserver zu verwenden. Wenn Sie keine Verschlüsselung während der Übertragung für die REST-Punkte wünschen, deaktivieren Sie diese Konfigurationen. Wir empfehlen Ihnen, die Standardkonfiguration für maximale Sicherheit zu verwenden.

Amazon EMR verwendet [security.ssl.algorithms](#), um die Liste der Chiffren anzugeben, die AES-basierte Verschlüsselung verwenden. Überschreiben Sie diese Konfiguration, um die gewünschten Chiffren zu verwenden.

Weitere Informationen finden Sie unter [SSL-Setup](#) in der Flink-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Flink History Server	historyserver.web.port	8082	TLS	emr-7.3.0+
Job Manager Restserver	rest.bind-port rest.port	Dynamisch	TLS	emr-7.3.0+

HBase

Amazon EMR setzt [Secure Hadoop RPC auf](#) `privacy`. HMaster und RegionServer verwenden Sie eine SASL-basierte Verschlüsselung bei der Übertragung. Dies setzt voraus, dass die Kerberos-Authentifizierung in der Sicherheitskonfiguration aktiviert ist.

Amazon EMR ist `hbase.ssl.enabled` auf `true` gesetzt und verwendet TLS für UI-Endpunkte. Wenn Sie TLS nicht für UI-Endpunkte verwenden möchten, deaktivieren Sie diese Konfiguration. Wir empfehlen, die Standardkonfiguration für maximale Sicherheit zu verwenden.

Amazon EMR setzt `hbase.rest.ssl.enabled` `hbase.thrift.ssl.enabled` und verwendet TLS für die REST- bzw. Thirft-Serverendpunkte. Wenn Sie TLS für diese Endpunkte nicht verwenden möchten, deaktivieren Sie diese Konfiguration. Wir empfehlen, die Standardkonfiguration für maximale Sicherheit zu verwenden.

Ab EMR 7.6.0 wird TLS auf HMaster und RegionServer Endpunkten unterstützt. Amazon EMR legt auch `hbase.server.netty.tls.enabled` und `hbase.client.netty.tls.enabled` fest. Wenn Sie TLS für diese Endpunkte nicht verwenden möchten, deaktivieren Sie diese Konfiguration. Wir empfehlen, die Standardkonfiguration zu verwenden, die Verschlüsselung und damit höhere Sicherheit bietet. Weitere Informationen finden Sie unter [Transport Level Security \(TLS\) bei der HBase RPC-Kommunikation](#) im Apache HBase Reference Guide.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
HMaster	HMaster	16000	SASL + Kerberos TLS	SASL + Kerberos in emr-4.8.0+, emr-5.0.0+, emr-6.0.0+ und emr-7.0.0+ TLS unter emr-7.6.0+
HMaster	HMaster UI	16010	TLS	emr-7.3.0+

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
RegionServer	RegionServer	16020	SASL + Kerberos TLS	SASL + Kerberos in emr-4.8.0+, emr-5.0.0+, emr-6.0.0+ und emr-7.0.0+ TLS unter emr-7.6.0+
RegionServer	RegionServer Informationen	16030	TLS	emr-7.3.0+
HBase Server ausruhen	Rest-Server	8070	TLS	emr-7.3.0+
HBase Server ausruhen	Rest-Benutzeroberfläche	8085	TLS	emr-7.3.0+
Hbase Thrift Server	Sparsamer Server	9090	TLS	emr-7.3.0+
Hbase Thrift Server	Thrift Server-Benutzeroberfläche	9095	TLS	emr-7.3.0+

Phoenix

Wenn Sie die Verschlüsselung während der Übertragung in Ihrem EMR-Cluster aktiviert haben, unterstützt Phoenix Query Server die TLS-Eigenschaft `phoenix.queryserver.tls.enabled`, die standardmäßig auf `true` gesetzt ist.

Weitere Informationen finden Sie unter [Konfigurationen in Bezug auf HTTPS](#) in der Phoenix Query Server-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Server abfragen	phoenix.queryserver.http.port	8765	TLS	emr-7.3.0+

Oozie

[OOZIE-3673](#) ist auf Amazon EMR verfügbar, wenn Sie Oozie auf Amazon EMR 7.3.0 und höher ausführen. Wenn Sie beim Ausführen einer E-Mail-Aktion benutzerdefinierte SSL- oder TLS-Protokolle konfigurieren müssen, können Sie die Eigenschaft in der Datei festlegen. `oozie.email.smtp.ssl.protocols oozie-site.xml` Wenn Sie die Verschlüsselung bei der Übertragung aktiviert haben, verwendet Amazon EMR standardmäßig das Protokoll TLS v1.3.

[OOZIE-3677](#) und [OOZIE-3674](#) sind auch auf Amazon EMR verfügbar, wenn Sie Oozie auf Amazon EMR 7.3.0 und höher ausführen. `keyStoreType` Auf diese Weise können Sie `trustStoreType` die `oozie-site.xml` Eigenschaften und in angeben. OOZIE-3674 fügt den Parameter dem Oozie-Client hinzu `--insecure`, sodass dieser Zertifikatsfehler ignorieren kann.

Oozie erzwingt die Überprüfung des TLS-Hostnamens, was bedeutet, dass jedes Zertifikat, das Sie für die Verschlüsselung während der Übertragung verwenden, die Anforderungen zur Überprüfung des Hostnamens erfüllen muss. Wenn das Zertifikat die Kriterien nicht erfüllt, kann es sein, dass der Cluster in der `oozie share lib update` Phase, in der Amazon EMR den Cluster bereitstellt, hängen bleibt. Wir empfehlen Ihnen, Ihre Zertifikate zu aktualisieren, um sicherzustellen, dass sie der Hostnamenverifizierung entsprechen. Wenn Sie die Zertifikate jedoch nicht aktualisieren können,

können Sie SSL für Oozie deaktivieren, indem Sie die `oozie.https.enabled` Eigenschaft `false` in der Clusterkonfiguration auf setzen.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
EmbeddedOozieServer	oozie.https.port	11443	TLS	emr-7.3.0+
EmbeddedOozieServer	oozie.email.smtp.port	25	TLS	emr-7.3.0+

Hue

Standardmäßig unterstützt Hue TLS, wenn die Verschlüsselung während der Übertragung in Amazon EMR-Clustern aktiviert ist. Weitere Informationen zu Hue-Konfigurationen finden [Sie unter Hue mit HTTPS/SSL konfigurieren](#).

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Hue	http_port	8888	TLS	emr-7.4.0+

Livy

Standardmäßig unterstützt Livy TLS, wenn die Verschlüsselung während der Übertragung in Amazon EMR-Clustern aktiviert ist. Weitere Informationen zu Livy-Konfigurationen finden Sie unter [HTTPS mit Apache Livy aktivieren](#).

Ab Amazon EMR 7.3.0 können Sie, wenn Sie Verschlüsselung bei der Übertragung und Kerberos-Authentifizierung verwenden, den Livy-Server nicht für Spark-Anwendungen verwenden, die vom

Hive-Metastore abhängen. Dieses Problem wurde in [HIVE-16340](#) behoben und in SPARK-44114 vollständig behoben, wenn die [Open-Source-Spark-Anwendung](#) auf Hive 3 aktualisiert werden kann. In der Zwischenzeit können Sie dieses Problem umgehen, wenn Sie dies einrichten. `hive.metastore.use.SSL false` Weitere Informationen finden Sie unter [Konfigurieren von Anwendungen](#).

Weitere Informationen finden Sie unter [HTTPS mit Apache Livy aktivieren](#).

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Livy-Server	livy.server.port	8998	TLS	emr-7.4.0+

JupyterEnterpriseGateway

Standardmäßig unterstützt Jupyter Enterprise Gateway TLS, wenn die Verschlüsselung während der Übertragung in Amazon EMR-Clustern aktiviert ist. [Weitere Informationen zu den Jupyter Enterprise Gateway-Konfigurationen finden Sie unter Securing Enterprise Gateway Server](#).

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
jupyter_enterprise_gateway	c. Anschluss EnterpriseGatewayApp	9547	TLS	emr-7.4.0+

JupyterHub

JupyterHub Unterstützt standardmäßig TLS, wenn die Verschlüsselung während der Übertragung in Amazon EMR-Clustern aktiviert ist. Weitere Informationen finden Sie in der Dokumentation unter

[Aktivieren der JupyterHub SSL-Verschlüsselung](#). Es wird nicht empfohlen, die Verschlüsselung zu deaktivieren.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
jupyter_hub	c.-Anschluss JupyterHub	9443	TLS	emr-5.14.0+, emr-6.0.0+, emr-7.0.0+

Zeppelin

Standardmäßig unterstützt Zeppelin TLS, wenn Sie die Verschlüsselung während der Übertragung in Ihrem EMR-Cluster aktivieren. Weitere Informationen zu den Zeppelin-Konfigurationen finden Sie unter [SSL-Konfiguration](#) in der Zeppelin-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Zeppelin	zeppelin. server.ssl.port	8890	TLS	7.3.0+

Zookeeper

Amazon EMR legt fest `serverCnxnFactory`, dass `org.apache.zookeeper.server.NettyServerCnxnFactory` TLS für das Zookeeper-Quorum und die Client-Kommunikation aktiviert wird.

`secureClientPort` gibt den Port an, der TLS-Verbindungen abhört. Wenn der Client keine TLS-Verbindungen zu Zookeeper unterstützt, können sich Clients mit dem unsicheren Port 2181

verbinden, der unter angegeben ist. `clientPort` Sie können diese beiden Ports überschreiben oder deaktivieren.

Amazon EMR legt `sslQuorum` sowohl als auch `admin.forceHttps`, `true` um die TLS-Kommunikation für das Quorum und den Admin-Server zu aktivieren. Wenn Sie keine Verschlüsselung während der Übertragung für das Quorum und den Admin-Server wünschen, können Sie diese Konfigurationen deaktivieren. Wir empfehlen, die Standardkonfigurationen für maximale Sicherheit zu verwenden.

Weitere Informationen finden Sie unter [Verschlüsselung, Authentifizierung und Autorisierungsoptionen](#) in der Zookeeper-Dokumentation.

Komponente	Endpunkt	Port	Verschlüsselungsmechanismus bei der Übertragung	Ab Version unterstützt
Zookeeper-Server	<code>secureClientPort</code>	2281	TLS	emr-7.4.0+
Zookeeper-Server	Quorum-Ports	Es gibt 2: Follower verwenden 2888, um sich mit dem Anführer zu verbinden. Bei der Wahl des Führers werden 3888 verwendet	TLS	emr-7.4.0+
Zookeeper-Server	Admin.Server-Port	8341	TLS	emr-7.4.0+

AWS Identity and Access Management für Amazon EMR

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon-EMR-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Funktionsweise von Amazon EMR mit IAM](#)
- [Schritte für Laufzeit-Rollen für Amazon EMR](#)
- [Konfigurieren Sie IAM-Servicerollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen](#)
- [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon EMR ausführen.

Service-Benutzer - wenn Sie den Amazon-EMR-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie zur Ausführung von Aufgaben weitere Amazon-EMR-Features verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf ein Feature in Amazon EMR nicht zugreifen können, siehe [Fehlerbehebung für Amazon-EMR-Identität und -Zugriff](#).

Service-Administrator – Wenn Sie in Ihrem Unternehmen für die Amazon-EMR-Ressourcen zuständig sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon EMR. Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon-EMR-Feature und -Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die

Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Amazon EMR verwenden kann, finden Sie unter [Funktionsweise von Amazon EMR mit IAM](#).

IAM-Administrator – Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht Details darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon EMR erstellen können. Beispiele für identitätsbasierte Amazon-EMR-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#).

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode für die Selbstsignierung von Anforderungen finden Sie unter [AWS Signature Version 4 für API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [AWS Multi-Faktor-Authentifizierung \(MFA\) in IAM](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges](#)

Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb von Ihrem AWS-Konto, die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Methoden für die Übernahme einer Rolle](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.

- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.
- **Serviceübergreifender Zugriff** — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon ausgeführte Anwendungen EC2** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein

Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden](#).

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter

[Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer Inline-Richtlinie wählen, finden Sie unter [Auswählen zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter [Richtlinien zur Servicesteuerung](#) im AWS Organizations Benutzerhandbuch.
- **Ressourcenkontrollrichtlinien (RCPs)** — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter [Resource Control Policies \(RCPs\)](#) im AWS Organizations Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und

der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

Funktionsweise von Amazon EMR mit IAM

Bevor Sie mit IAM den Zugriff auf Amazon EMR verwalten können, sollten Sie sich darüber informieren, welche IAM-Features Sie mit Amazon EMR verwenden können.

IAM-Features, die Sie mit Amazon EMR verwenden können

IAM-Feature	Amazon-EMR-Support
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie Amazon EMR und andere AWS Services mit den meisten IAM-Funktionen funktionieren, finden Sie im [AWS IAM-Benutzerhandbuch unter Services, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für Amazon EMR

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon EMR

Beispiele für identitätsbasierte Amazon-EMR-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#).

Ressourcenbasierte Richtlinien in Amazon EMR

Unterstützt ressourcenbasierte Richtlinien: Ja

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für Amazon EMR

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste von Amazon EMR Aktionen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EMR](#) in der Service-Autorisierungs-Referenz.

Richtlinienaktionen in Amazon EMR verwenden das folgende Präfix vor der Aktion:

```
EMR
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
```

```
"EMR:action1",  
"EMR:action2"  
]
```

Beispiele für identitätsbasierte Amazon-EMR-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#).

Richtlinienressourcen für Amazon EMR

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der Amazon EMR-Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon EMR definierte Ressourcen](#) in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN jeder Ressource angeben können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EMR](#).

Beispiele für identitätsbasierte Amazon-EMR-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#).

Richtlinien-Bedingungsschlüssel für Amazon EMR

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der Bedingungsschlüssel von Amazon EMR und Informationen darüber, welche Aktionen und Ressourcen Sie mit einem Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EMR](#) in der Service-Autorisierungs-Referenz.

Beispiele für identitätsbasierte Amazon-EMR-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#).

Zugriffskontrolllisten (ACLs) in Amazon EMR

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffssteuerung (ABAC) mit Amazon EMR

Unterstützt ABAC (Tags in Richtlinien)	Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement 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 temporärer Anmeldeinformationen mit Amazon EMR

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen,

werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln von einer Benutzerrolle zu einer IAM-Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

Serviceübergreifende Prinzipalberechtigungen für Amazon EMR

Unterstützt Forward Access Sessions (FAS): Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für Amazon EMR

Unterstützt Servicerollen	Nein

Serviceverknüpfte Rollen für Amazon EMR

Unterstützt serviceverknüpfte Rollen	Ja

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Verwenden Sie Cluster- und Notebook-Tags mit IAM-Richtlinien für die Zugriffskontrolle

Die Berechtigungen für Amazon-EMR-Aktionen, die EMR Notebooks und EMR-Clustern zugeordnet sind, können anhand der Tag-basierten Zugriffskontrolle mit identitätsbasierten IAM-Richtlinien abgestimmt werden. Sie können Bedingungsschlüssel in einem Condition-Element (auch als Condition-Block bezeichnet) verwenden, um bestimmte Aktionen nur dann zuzulassen, wenn ein Notebook, ein Cluster oder beide bestimmte Tag-Schlüssel oder Schlüssel-Wert-Kombinationen aufweisen. Sie können auch die Aktion `CreateEditor` (erstellt ein EMR Notebook) und die Aktion `RunJobFlow` (erstellt einen Cluster) einschränken, damit bei der Erstellung der Ressource eine Anforderung für ein Tag eingereicht werden muss.

In Amazon EMR gelten die Bedingungsschlüssel, die in einem Condition-Element verwendet werden können, nur für die Amazon-EMR-API-Aktionen, für die `ClusterID` oder `NotebookID` ein erforderlicher Anforderungsparameter ist. Beispielsweise unterstützt die [ModifyInstanceGroups](#)Aktion keine Kontextschlüssel, da es sich um einen optionalen Parameter `ClusterID` handelt.

Wenn Sie ein EMR-Notebook erstellen, wird ein Standard-Tag angewendet. Dabei entspricht die Schlüsselzeichenfolge `creatorUserId` der IAM-Benutzer-ID, mit der das Notebook erstellt wurde. Dies ist nützlich, um zulässige Aktionen für das Notebook ausschließlich auf den Ersteller zu beschränken.

Die folgenden Bedingungsschlüssel sind in Amazon EMR: verfügbar:

- Verwenden Sie den Bedingungskontextschlüssel `elasticmapreduce:ResourceTag/TagKeyString`, um Benutzeraktionen in Clustern oder Notebooks mit Tags mit dem von Ihnen festgelegten `TagKeyString` zuzulassen oder abzulehnen. Wenn eine Aktion sowohl `NotebookID` als auch `ClusterID` übergibt, gilt die Bedingung sowohl für den Cluster als auch das Notebook. Das bedeutet, dass beide Ressourcen dieselbe Tag-Schlüsselzeichenfolge oder Schlüssel-Wert-Kombination aufweisen müssen. Sie können das Element `Resource` verwenden, um die Anweisung nach Bedarf nur auf Cluster oder Notebooks zu beschränken. Weitere Informationen finden Sie unter [Beispiele für identitätsbasierte Amazon-EMR-Richtlinien](#).
- Verwenden Sie den `elasticmapreduce:RequestTag/TagKeyString` Bedingungskontextschlüssel, um bei actions/API Aufrufen ein bestimmtes Tag vorzuschreiben. Verwenden Sie diesen Bedingungskontextschlüssel zusammen mit der `CreateEditor`-Aktion, um festzulegen, dass bei der Erstellung von Notebooks ein Schlüssel mit `TagKeyString` angewendet wird.

Beispiele

Eine Liste der Amazon EMR Aktionen finden Sie unter [Von Amazon EMR definierte Aktionen](#) im IAM-Benutzerhandbuch.

Schritte für Laufzeit-Rollen für Amazon EMR

Eine Runtime-Rolle ist eine AWS Identity and Access Management (IAM) -Rolle, die Sie angeben können, wenn Sie einen Job oder eine Abfrage an einen Amazon EMR-Cluster senden. Der Job oder die Abfrage, die Sie an Ihren Amazon EMR-Cluster senden, verwendet die Runtime-Rolle, um auf AWS Ressourcen wie Objekte in Amazon S3 zuzugreifen. Sie können Laufzeit-Rollen mit Amazon EMR für Spark- und Hive-Jobs angeben.

Sie können auch Laufzeit-Rollen angeben, wenn Sie eine Verbindung zu Amazon-EMR-Clustern in einem EMR-Cluster herstellen Amazon SageMaker AI und wenn Sie einen Amazon EMR Studio Workspace an einen EMR-Cluster anhängen. Weitere Informationen finden Sie unter [Connect zu einem Amazon EMR-Cluster von SageMaker AI Studio](#) herstellen und [Einen EMR-Studio-Workspace mit einer Laufzeit-Rolle ausführen](#).

Zuvor führten Amazon-EMR-Cluster Amazon-EMR-Aufträge oder -Abfragen mit Berechtigungen aus, die auf der IAM-Richtlinie basierten, die dem Instance-Profil zugeordnet war, das Sie zum Starten des Clusters verwendet haben. Das bedeutete, dass die Richtlinien die Vereinigung aller Berechtigungen für alle Aufträge und Abfragen enthalten mussten, die auf einem Amazon-EMR-Cluster ausgeführt wurden. Mit Laufzeit-Rollen können Sie jetzt die Zugriffskontrolle für jeden Auftrag oder jede Abfrage einzeln verwalten, anstatt das Amazon-EMR-Instance-Profil des Clusters gemeinsam zu nutzen.

Auf Amazon EMR-Clustern mit Runtime-Rollen können Sie auch eine AWS Lake Formation basierte Zugriffskontrolle auf Spark-, Hive- und Presto-Jobs und Abfragen für Ihre Data Lakes anwenden. Weitere Informationen zur Integration mit AWS Lake Formation finden Sie unter [Integrieren Sie Amazon EMR mit AWS Lake Formation](#)

Note

Wenn Sie eine Runtime-Rolle für einen Amazon EMR-Schritt angeben, können die Jobs oder Abfragen, die Sie einreichen, nur auf AWS Ressourcen zugreifen, die die mit der Runtime-Rolle verknüpften Richtlinien zulassen. Diese Jobs und Abfragen können nicht auf den Instance-Metadaten-Service auf den EC2 Instances des Clusters zugreifen oder das EC2 Instance-Profil des Clusters für den Zugriff auf AWS Ressourcen verwenden.

Voraussetzungen für den Start eines Amazon-EMR-Clusters mit einer Laufzeit-Rolle

Themen

- [Schritt 1: Sicherheitskonfigurationen in Amazon EMR einrichten](#)
- [Schritt 2: Richten Sie ein EC2 Instance-Profil für den Amazon EMR-Cluster ein](#)
- [Schritt 3: Eine Vertrauensrichtlinie einrichten](#)

Schritt 1: Sicherheitskonfigurationen in Amazon EMR einrichten

Verwenden Sie die folgende JSON-Struktur, um eine Sicherheitskonfiguration für AWS Command Line Interface (AWS CLI) zu erstellen, und legen Sie `EnableApplicationScopedIAMRole` sie auf `true`. Weitere Informationen zu den Sicherheitskonfigurationen finden Sie unter [Verwenden Sie Sicherheitskonfigurationen, um die Amazon EMR-Cluster-Sicherheit einzurichten](#).

```
{
  "AuthorizationConfiguration":{
    "IAMConfiguration":{
      "EnableApplicationScopedIAMRole":true
    }
  }
}
```

Wir empfehlen, in der Sicherheitskonfiguration immer die Verschlüsselungsoptionen bei der Übertragung zu aktivieren, sodass Daten, die über das Internet übertragen werden, verschlüsselt und nicht im Klartext übertragen werden. Sie können diese Optionen überspringen, wenn Sie keine Verbindung zu Amazon EMR-Clustern mit Runtime-Rollen aus SageMaker Runtime Studio oder EMR Studio herstellen möchten. Informationen zur Konfiguration der Datenverschlüsselung finden Sie unter [Datenverschlüsselung konfigurieren](#).

Alternativ können Sie mit dem eine Sicherheitskonfiguration mit benutzerdefinierten Einstellungen mit [AWS Management Console](#) erstellen.

Schritt 2: Richten Sie ein EC2 Instance-Profil für den Amazon EMR-Cluster ein

Amazon EMR-Cluster verwenden die EC2 Amazon-Instance-Profilrolle, um die Runtime-Rollen zu übernehmen. Um Laufzeit-Rollen mit Amazon-EMR-Schritten zu verwenden, fügen Sie der IAM-Rolle, die Sie als Instance-Profilrolle verwenden möchten, die folgenden Richtlinien hinzu. Informationen zum Hinzufügen von Richtlinien zu einer IAM-Rolle oder zum Bearbeiten einer

vorhandenen Inline- oder verwalteten Richtlinie finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRuntimeRoleUsage",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Resource": [
        <runtime-role-ARN>
      ]
    }
  ]
}
```

Schritt 3: Eine Vertrauensrichtlinie einrichten

Legen Sie für jede IAM-Rolle, die Sie als Laufzeit-Rolle verwenden möchten, die folgende Vertrauensrichtlinie fest und ersetzen Sie sie `EMR_EC2_DefaultRole` durch Ihre Instance-Profilrolle. Informationen zum Ändern der Vertrauensrichtlinie einer IAM-Rolle finden Sie unter [Vertrauensrichtlinie für Rollen ändern](#).

```
{
  "Sid": "AllowAssumeRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
  },
  "Action": "sts:AssumeRole"
}
```

Starten Sie einen Amazon-EMR-Cluster mit rollenbasierter Zugriffskontrolle

Nachdem Sie Ihre Konfigurationen eingerichtet haben, können Sie einen Amazon-EMR-Cluster mit der Sicherheitskonfiguration von [Schritt 1: Sicherheitskonfigurationen in Amazon EMR einrichten](#) starten. Um Runtime-Rollen mit Amazon EMR-Schritten zu verwenden, verwenden Sie Release

Label `emr-6.7.0` oder höher und wählen Sie Hive, Spark oder beide als Cluster-Anwendung aus. CloudWatchAgent wird auf Runtime Role Clusters für EMR 7.6 und höher unterstützt. Um von SageMaker AI Studio aus eine Verbindung herzustellen, verwenden Sie Release `emr-6.9.0` oder höher und wählen Sie Livy, Spark, Hive oder Presto als Ihre Cluster-Anwendung aus. Anweisungen zum Start Ihres Clusters finden Sie unter [Geben Sie eine Sicherheitskonfiguration für einen Amazon EMR-Cluster an](#).

Spark-Jobs mithilfe der Amazon-EMR-Schritte übermitteln

Im Folgenden finden Sie ein Beispiel für die Ausführung des in Apache Spark enthaltenen `HdfsTest` Beispiels. Dieser API-Aufruf ist nur erfolgreich, wenn die bereitgestellte Amazon-EMR-Laufzeitrolle auf die `S3_LOCATION` zugreifen kann.

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
S3_LOCATION=<s3-path>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>

aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Spark Example", "ActionOnFailure": "CONTINUE", "HadoopJarStep":
{ "Jar": "command-runner.jar", "Args" : ["spark-example", "HdfsTest",
"$S3_LOCATION"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

Note

Wir empfehlen, den SSH-Zugriff auf den Amazon-EMR-Cluster zu deaktivieren und nur der Amazon-EMR-AddJobFlowSteps-API den Zugriff auf den Cluster zu gewähren.

Hive-Jobs mithilfe der Amazon-EMR-Schritte übermitteln

Im folgenden Beispiel wird Apache Hive mit Amazon-EMR-Schritten verwendet, um einen Job zur Ausführung der `QUERY_FILE.hql` Datei einzureichen. Diese Abfrage ist nur erfolgreich, wenn die angegebene Laufzeit-Rolle auf den Amazon-S3-Pfad der Abfragedatei zugreifen kann.

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>
```

```
aws emr add-steps --cluster-id $CLUSTER_ID \  
--steps '[{"Name": "Run hive query using command-runner.jar - simple  
select", "ActionOnFailure": "CONTINUE", "HadoopJarStep": { "Jar": "command-  
runner.jar", "Args" : ["hive -  
f", "s3://DOC_EXAMPLE_BUCKET/QUERY_FILE.hql"] } }]' \  
--execution-role-arn $RUNTIME_ROLE_ARN \  
--region $REGION
```

Stellen Sie über ein SageMaker AI Studio-Notizbuch eine Connect zu Amazon EMR-Clustern mit Runtime-Rollen her

Sie können Amazon EMR-Laufzeitrollen auf Abfragen anwenden, die Sie in Amazon EMR-Clustern von SageMaker AI Studio aus ausführen. Führen Sie dazu die folgenden Schritte aus.

1. Folgen Sie den Anweisungen unter [Amazon SageMaker AI Studio starten](#), um ein SageMaker AI Studio zu erstellen.
2. Starten Sie in der SageMaker AI Studio-Benutzeroberfläche ein Notebook mit unterstützten Kernen. Starten Sie beispielsweise ein SparkMagic Image mit einem PySpark Kernel.
3. Wählen Sie in SageMaker AI Studio einen Amazon EMR-Cluster und dann Connect aus.
4. Wählen Sie eine Laufzeit-Rolle und dann Verbinden aus.

Dadurch wird eine SageMaker KI-Notebookzelle mit magischen Befehlen erstellt, um eine Verbindung zu Ihrem Amazon EMR-Cluster mit der ausgewählten Amazon EMR-Laufzeitrolle herzustellen. In der Notebook-Zelle können Sie Abfragen mit Laufzeit-Rollen- und Lake-Formation-basierter Zugriffskontrolle eingeben und ausführen. Ein detaillierteres Beispiel finden Sie unter [Anwenden detaillierter Datenzugriffskontrollen mit AWS Lake Formation und Amazon EMR von Amazon SageMaker AI Studio aus](#).

Steuern Sie den Zugriff auf die Amazon-EMR-Laufzeit-Rolle

Sie können den Zugriff auf die Laufzeit-Rolle mit dem Bedingungsschlüssel `elasticmapreduce:ExecutionRoleArn` steuern. Die folgende Richtlinie ermöglicht es einem IAM-Prinzipal, eine IAM-Rolle mit dem Namen `Caller` oder eine beliebige IAM-Rolle, die mit der Zeichenfolge `CallerTeamRole` beginnt, als Laufzeitrolle zu verwenden.

Important

Sie müssen eine auf dem `elasticmapreduce:ExecutionRoleArn` Kontextschlüssel basierende Bedingung erstellen, wenn Sie einem Anrufer Zugriff zum Aufrufen von

AddJobFlowSteps oder gewähren GetClusterSessionCredentials APIs, wie das folgende Beispiel zeigt.

```
{
  "Sid": "AddStepsWithSpecificExecRoleArn",
  "Effect": "Allow",
  "Action": [
    "elasticmapreduce:AddJobFlowSteps"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ExecutionRoleArn": [
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/Caller"
      ]
    },
    "StringLike": {
      "elasticmapreduce:ExecutionRoleArn": [
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/CallerTeamRole*"
      ]
    }
  }
}
```

Schaffen Sie Vertrauen zwischen Laufzeit-Rollen und Amazon-EMR-Clustern

Amazon EMR generiert eine eindeutige Kennung ExternalId für jede Sicherheitskonfiguration mit aktivierter Laufzeit-Rollenautorisierung. Diese Autorisierung ermöglicht es jedem Benutzer, eine Reihe von Laufzeit-Rollen zu besitzen, die er auf Clustern verwenden kann, die ihm gehören. In einem Unternehmen kann beispielsweise jede Abteilung ihre externe ID verwenden, um die Vertrauensrichtlinie für ihre eigenen Laufzeit-Rollen zu aktualisieren.

Sie können die externe ID mit der Amazon-EMR-DescribeSecurityConfiguration-API finden, wie im folgenden Beispiel gezeigt.

```
aws emr describe-security-configuration --name 'iamconfig-with-lf' {"Name": "iamconfig-with-lf",
  "SecurityConfiguration": {
    {"AuthorizationConfiguration": {"IAMConfiguration": {
      {"EnableApplicationScopedIAMRole\
```

```

      ":true,\"ApplicationScopedIAMRoleConfiguration\":{\"PropagateSourceIdentity
\":true,\"ExternalId\":{\"FXH5TSACFDWUCDSR3YQE207ETPUSM40BCGLYW0DSCUZNZ4Y\"}},\"Lake
FormationConfiguration\":{\"AuthorizedSessionTagValue\":{\"Amazon EMR\"}}}\",
      \"CreationDateTime\": \"2022-06-03T12:52:35.308000-07:00\"
    }
  }
}

```

Informationen zur Verwendung einer externen ID finden Sie unter [So verwenden Sie eine externe ID, wenn Sie einem Dritten Zugriff auf Ihre AWS Ressourcen gewähren](#).

Audit

Um die Aktionen zu überwachen und zu kontrollieren, die Endbenutzer mit IAM-Rollen ergreifen, können Sie das Quellidentitätsfeature aktivieren. Weitere Informationen zur Quellenidentität finden Sie unter [Überwachen und Steuern von Aktionen mit übernommenen Rollen](#).

Um die Quellidentität nachzuverfolgen, stellen Sie `ApplicationScopedIAMRoleConfiguration/PropagateSourceIdentity` in Ihrer Sicherheitskonfiguration wie folgt auf `true` ein.

```

{
  "AuthorizationConfiguration":{
    "IAMConfiguration":{
      "EnableApplicationScopedIAMRole":true,
      "ApplicationScopedIAMRoleConfiguration":{
        "PropagateSourceIdentity":true
      }
    }
  }
}

```

Wenn Sie `PropagateSourceIdentity` auf `true` einstellen, wendet Amazon EMR die Quellidentität aus den Anrufermeldedaten auf einen Job oder eine Abfragesitzung an, die Sie mit der Laufzeit-Rolle erstellen. Wenn in den Anrufermeldedaten keine Quellidentität enthalten ist, legt Amazon EMR die Quellidentität nicht fest.

Um diese Eigenschaft zu verwenden, geben Sie wie folgt `sts:SetSourceIdentity`-Berechtigungen für Ihr Instance-Profil ein.

```

{ // PropagateSourceIdentity statement
  "Sid":"PropagateSourceIdentity",
  "Effect":"Allow",

```

```

    "Action": "sts:SetSourceIdentity",
    "Resource": [
        <runtime-role-ARN>
    ],
    "Condition": {
        "StringEquals": {
            "sts:SourceIdentity": <source-identity>
        }
    }
}

```

Sie müssen die AllowSetSourceIdentity-Anweisung auch zur Vertrauensrichtlinie Ihrer Laufzeit-Rollen hinzufügen.

```

{ // AllowSetSourceIdentity statement
  "Sid": "AllowSetSourceIdentity",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
  },
  "Action": [
    "sts:SetSourceIdentity",
    "sts:AssumeRole"
  ],
  "Condition": {
    "StringEquals": {
      "sts:SourceIdentity": <source-identity>
    }
  }
}

```

Weitere Überlegungen

Note

Mit der Amazon EMR-Version kann es zu zeitweiligen Ausfällen kommen `emr-6.9.0`, wenn Sie von AI Studio aus SageMaker eine Verbindung zu Amazon EMR-Clustern herstellen. Um dieses Problem zu lösen, können Sie den Patch mit einer Bootstrap-Aktion installieren, wenn Sie den Cluster starten. Einzelheiten zum Patch finden Sie unter [Bekannte Probleme in Amazon EMR Version 6.9.0](#).

Beachten Sie außerdem Folgendes, wenn Sie Laufzeit-Rollen für Amazon EMR konfigurieren.

- Amazon EMR unterstützt Laufzeit-Rollen in allen kommerziellen AWS-Regionen Anwendungen.
- Amazon-EMR-Schritte unterstützen Apache Spark- und Apache Hive-Jobs mit Laufzeit-Rollen, wenn Sie Release `emr-6.7.0` oder höher verwenden.
- SageMaker AI Studio unterstützt Spark-, Hive- und Presto-Abfragen mit Runtime-Rollen, wenn Sie Release oder höher verwenden. `emr-6.9.0`
- Die folgenden Notebook-Kernel in SageMaker AI unterstützen Runtime-Rollen:
 - DataScience — Python-3-Kernel
 - DataScience 2.0 — Python-3-Kernel
 - DataScience 3.0 — Python-3-Kernel
 - SparkAnalytics 1.0 — SparkMagic und PySpark Kernel
 - SparkAnalytics 2.0 — SparkMagic und Kernel PySpark
 - SparkMagic — Kernel PySpark
- Amazon EMR unterstützt Schritte, die RunJobFlow nur zum Zeitpunkt der Clustererstellung verwendet werden. Diese API unterstützt keine Laufzeit-Rollen.
- Amazon EMR unterstützt keine Laufzeit-Rollen auf Clustern, die Sie so konfigurieren, dass sie hochverfügbar sind.
- Ab Amazon EMR Version 7.5.0 und höher unterstützen Runtime-Rollen die Anzeige von Spark- und YARN-Benutzeroberflächen (UIs), wie z. B. die folgenden: Spark Live UI, Spark History Server NodeManager, YARN und YARN. ResourceManager Wenn Sie zu diesen wechseln UIs, werden Sie nach einem Benutzernamen und einem Passwort gefragt. Benutzernamen und Passwörter können mithilfe der `GetClusterSessionCredentials` EMR-API generiert werden. Weitere Informationen zur Nutzung der API finden Sie unter [GetClusterSessionCredentials](#)

Ein Beispiel für die Verwendung der `GetClusterSessionCredentials` EMR-API ist das Folgende:

```
aws emr get-cluster-session-credentials --cluster-id <cluster_ID> --execution-role-arn <IAM_role_arn>
```

- Sie müssen Ihre Bash-Befehlsargumente umgehen, wenn Sie Befehle mit der `command-runner.jar` JAR-Datei ausführen:

```
aws emr add-steps --cluster-id <cluster-id> --steps '[{"Name":"sample-step","ActionOnFailure":"CONTINUE","Jar":"command-runner.jar","Properties":"","Args":
```

```
["bash", "-c", "\"aws s3 ls\"", "Type": "CUSTOM_JAR"]]' --execution-role-  
arn <IAM_ROLE_ARN>
```

Außerdem müssen Sie Ihre Bash-Befehlsargumente maskieren, wenn Sie Befehle mit dem Script-Runner ausführen. Das folgende Beispiel zeigt die Einstellung von Spark-Eigenschaften mit enthaltenen Escape-Zeichen:

```
"\"--conf spark.sql.autoBroadcastJoinThreshold=-1\n--conf  
spark.cradle.RSv2Mode.enabled=true\""
```

- Laufzeit-Rollen bieten keine Unterstützung für die Steuerung des Zugriffs auf Cluster-Ressourcen wie HDFS und HMS.

Konfigurieren Sie IAM-Servicerollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen

Amazon EMR und Anwendungen wie Hadoop und Spark benötigen Berechtigungen für den Zugriff auf andere AWS -Ressourcen und die Ausführung von Aktionen während der Ausführung. Jeder Cluster in Amazon EMR muss eine Servicerolle und eine Rolle für das EC2 Amazon-Instance-Profil haben. Weitere Informationen finden Sie unter [IAM-Rollen](#) und [Verwenden von Instance-Profilen](#) im IAM-Benutzerhandbuch. Die diesen Rollen zugeordneten IAM-Richtlinien stellen dem Cluster die Berechtigung bereit, im Namen eines Benutzers mit anderen AWS -Services zu interagieren.

Eine zusätzliche Rolle, die Auto Scaling-Rolle, ist erforderlich, wenn Ihr Cluster ein Auto Scaling in Amazon EMR verwendet. Die AWS Servicerolle für EMR Notebooks ist erforderlich, wenn Sie EMR Notebooks verwenden.

Amazon EMR bietet Standardrollen und standardmäßig verwaltete Richtlinien für alle Rollen, die Berechtigungen bestimmen. Verwaltete Richtlinien werden von erstellt und verwaltet AWS, sodass sie automatisch aktualisiert werden, wenn sich die Serviceanforderungen ändern. Siehe unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

Wenn Sie zum ersten Mal einen Cluster oder ein Notebook in einem Konto erstellen, existieren für Amazon EMR noch keine Rollen. Nachdem Sie sie erstellt haben, können Sie die Rollen, die mit ihnen verknüpften Richtlinien und die Berechtigungen, die durch die Richtlinien zugelassen oder verweigert wurden, in der IAM-Konsole anzeigen (<https://console.aws.amazon.com/iam/>). Sie können Standardrollen für Amazon EMR zum Erstellen und Verwenden angeben, Sie können Ihre eigenen Rollen erstellen und sie einzeln angeben, wenn Sie einen Cluster erstellen, um Berechtigungen

anpassen. Und Sie können Standardrollen angeben, die verwendet werden sollen, wenn Sie einen Cluster mithilfe der AWS CLI erstellen. Weitere Informationen finden Sie unter [Passen Sie IAM-Rollen mit Amazon EMR an](#).

Ändern von identitätsbasierten Richtlinien für Berechtigungen zum Übergeben von Servicerollen für Amazon EMR

Die verwalteten Standardrichtlinien von Amazon EMR mit vollen Berechtigungen beinhalten `iam:PassRole`-Sicherheitskonfigurationen, darunter die folgenden:

- `iam:PassRole`-Berechtigungen nur für bestimmte Amazon-EMR-Standardrollen.
- `iam:PassedToService`Bedingungen, die es Ihnen ermöglichen, die Richtlinie nur mit bestimmten AWS Diensten zu verwenden, z. B. `elasticmapreduce.amazonaws.com` und `undec2.amazonaws.com`.

Sie können die JSON-Version der [Amazon EMRFull AccessPolicy_v2- und Amazon EMRServicePolicy_v2-Richtlinien in der IAM-Konsole](#) anzeigen. Wir empfehlen, dass Sie neue Cluster mit den verwalteten v2-Richtlinien erstellen.

Übersicht über Servicerollen

In der folgenden Tabelle finden Sie als Referenz die IAM-Servicerollen aufgelistet, die Amazon EMR zugeordnet sind.

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
Servicerolle für Amazon EMR (EMR-Rolle)	EMR_DefaultRole_V2	Ermöglicht Amazon EMR, in Ihrem Namen andere AWS Services aufzurufen, wenn Ressourcen bereitgestellt und Service-Level-Aktionen ausgeführt werden. Diese Rolle ist für alle Cluster erforderlich.	AmazonEMRServicePolicy_v2  Important Zum Anfordern von Spot Instances ist eine serviceve

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
			rknüpfte Rolle erforderlich. Wenn diese Rolle nicht vorhanden ist, benötigt die Amazon-EMR-Servicerolle die Berechtigung, sie zu erstellen, andernfalls tritt ein Berechtigungsfehler auf. Wenn Sie Spot Instances anfordern möchten, müssen Sie diese Richtlinie so aktualisieren, dass sie eine Erklärung enthält, die die Erstellung dieser serviceverknüpften Rolle ermöglicht. Weitere Informati

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
			onen finden Sie unter Servicerolle für Amazon EMR (EMR-Rolle) und serviceverknüpfte Rolle für Spot-Instance-Anfragen im EC2 Amazon-Benutzerhandbuch.

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
Servicerolle für EC2 Cluster-Instances (EC2Instance-Profil)	EMR_EC2_DefaultRole	Anwendungsprozesse, die auf dem Hadoop-Ökosystem auf Cluster-Instances ausgeführt werden, verwenden diese Rolle, wenn sie andere AWS Dienste aufrufen. Für den Zugriff auf Daten in Amazon S3 mithilfe von EMRFS können Sie unterschiedliche Rollen angeben, die abhängig vom Speicherort der Daten in Amazon S3 angenommen werden sollen. Beispielsweise können mehrere Teams auf ein einzelnes „Datenspeicherkonto“ von Amazon S3 zugreifen. Weitere Informationen finden Sie unter Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3 . Diese Rolle ist für alle Cluster erforderlich.	AmazonElasticMapReduceforEC2Role . Weitere Informationen finden Sie unter Servicerolle für EC2 Cluster-Instances (EC2Instance-Profil) .

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
Servicerolle für Auto Scaling in Amazon EMR (Auto Scaling-Rolle)	EMR_AutoScaling_DefaultRole	Ermöglicht zusätzliche Aktionen für dynamisch skalierte Umgebungen. Nur erforderlich für Cluster mit Auto Scaling in Amazon EMR. Weitere Informationen finden Sie unter Verwenden der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instanzgruppen in Amazon EMR .	AmazonElasticMapReduceforAutoScalingRole . Weitere Informationen finden Sie unter Servicerolle für Auto Scaling in Amazon EMR (Auto Scaling-Rolle) .

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
Servicerolle für EMR Notebooks	EMR_Notebooks_DefaultRole	Stellt Berechtigungen bereit, die ein EMR-Notizbuch benötigt, um auf andere AWS Ressourcen zuzugreifen und Aktionen auszuführen. Nur erforderlich, wenn EMR-Notebooks verwendet werden.	AmazonElasticMapReduceEditorsRole . Weitere Informationen finden Sie unter Servicerolle für EMR Notebooks. S3FullAccessPolicy wird auch standardmäßig angehängt. Im Folgenden finden Sie den Inhalt dieser Richtlinie.. <pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:*", "Resource": "*" }] }</pre>

Funktion	Standardrolle	Beschreibung	Verwaltete Standardrichtlinie
Serviceverknüpfte Rolle	AWSServiceRoleForEMRCleanup	Amazon EMR erstellt automatisch eine serviceverknüpfte Rolle. Wenn der Service für Amazon EMR nicht mehr in der Lage ist, EC2 Amazon-Ressourcen zu bereinigen, kann Amazon EMR diese Rolle zum Aufräumen verwenden. Wenn ein Cluster Spot-Instances verwendet, muss die Berechtigungsrichtlinie, die der Servicerolle für Amazon EMR (EMR-Rolle) angefügt ist, die Erstellung einer serviceverknüpften Rolle zulassen. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon EMR .	AmazonEMRCleanupPolicy

Themen

- [Von Amazon EMR verwendete IAM-Servicerollen](#)
- [Passen Sie IAM-Rollen mit Amazon EMR an](#)

- [Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3](#)
- [Ressourcenbasierte Richtlinien für den Zugriff von Amazon EMR auf AWS Glue Data Catalog verwenden](#)
- [IAM-Rollen mit Anwendungen verwenden, die AWS -Services direkt aufrufen](#)
- [Benutzern und Gruppen gestatten, Rollen zu erstellen und zu ändern](#)

Von Amazon EMR verwendete IAM-Servicerollen

Amazon EMR verwendet IAM-Servicerollen, um in Ihrem Namen Aktionen bei der Bereitstellung von Cluster-Ressourcen, der Ausführung von Anwendungen, der dynamischen Skalierung von Ressourcen und der Erstellung und Ausführung von EMR Notebooks durchzuführen. Amazon EMR verwendet in Interaktionen mit anderen AWS -Services die folgenden Rollen. Jede Rolle verfügt über eine eindeutige Funktion innerhalb von Amazon EMR. Die Themen in diesem Abschnitt beschreiben die Rollenfunktion und stellen die Standardrollen und die Berechtigungsrichtlinie für jede Rolle bereit.

Wenn Ihr Cluster über Anwendungscode verfügt, der AWS Dienste direkt aufruft, müssen Sie möglicherweise das SDK verwenden, um Rollen anzugeben. Weitere Informationen finden Sie unter [IAM-Rollen mit Anwendungen verwenden, die AWS -Services direkt aufrufen](#).

Themen

- [Servicerolle für Amazon EMR \(EMR-Rolle\)](#)
- [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#)
- [Servicerolle für Auto Scaling in Amazon EMR \(Auto Scaling-Rolle\)](#)
- [Servicerolle für EMR Notebooks](#)
- [Verwenden von serviceverknüpften Rollen für Amazon EMR](#)

Servicerolle für Amazon EMR (EMR-Rolle)

Die Amazon EMR-Rolle definiert die zulässigen Aktionen für Amazon EMR, wenn es Ressourcen bereitstellt und Service-Level-Aufgaben ausführt, die nicht im Kontext einer EC2 Amazon-Instance ausgeführt werden, die innerhalb eines Clusters ausgeführt wird. Die Service-Rolle wird beispielsweise verwendet, um EC2 Instances bereitzustellen, wenn ein Cluster gestartet wird.

- Der Standardrollenname ist `EMR_DefaultRole_V2`.

- Die Amazon EMR angefügte standardmäßige verwaltete Richtlinie `EMR_DefaultRole_V2` ist `AmazonEMRServicePolicy_v2`. Diese v2-Richtlinie ersetzt die veraltete verwaltete Standardrichtlinie `AmazonElasticMapReduceRole`.

`AmazonEMRServicePolicy_v2` hängt vom begrenzten Zugriff auf Ressourcen ab, die Amazon EMR bereitstellt oder nutzt. Wenn Sie diese Richtlinie verwenden, müssen Sie bei der Bereitstellung des Clusters das Benutzer-Tag `for-use-with-amazon-emr-managed-policies = true` übergeben. Amazon EMR verbreitet diese Tags automatisch. Darüber hinaus müssen Sie möglicherweise manuell ein Benutzer-Tag zu bestimmten Ressourcentypen hinzufügen, z. B. EC2 Sicherheitsgruppen, die nicht von Amazon EMR erstellt wurden. Siehe [Taggen von Ressourcen zur Verwendung verwalteter Richtlinien](#).

 Important

Amazon EMR verwendet diese Amazon EMR-Servicerolle und die [AWSServiceRoleForEMRCleanup](#) Rolle, um Clusterressourcen in Ihrem Konto zu bereinigen, die Sie nicht mehr verwenden, z. B. EC2 Amazon-Instances. Sie müssen Aktionen für die Rollenrichtlinien angeben, um die Ressourcen zu löschen oder zu beenden. Andernfalls kann Amazon EMR diese Bereinigungsaktionen nicht durchführen, und es können Kosten für ungenutzte Ressourcen anfallen, die im Cluster verbleiben.

Im Folgenden werden die Inhalte der aktuellen `AmazonEMRServicePolicy_v2`-Richtlinie angezeigt. Sie können den aktuellen Inhalt der [AmazonEMRServicePolicy_v2](#) verwalteten Richtlinie auch auf der IAM-Konsole sehen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateInTaggedNetwork",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:RunInstances",
        "ec2:CreateFleet",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion"
      ],
    }
  ],
}
```

```

"Resource": [
  "arn:aws:ec2:*:*:subnet/*",
  "arn:aws:ec2:*:*:security-group/*"
],
"Condition": {
  "StringEquals": {
    "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
  }
},
{
  "Sid": "CreateWithEMRTaggedLaunchTemplate",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateFleet",
    "ec2:RunInstances",
    "ec2:CreateLaunchTemplateVersion"
  ],
  "Resource": "arn:aws:ec2:*:*:launch-template/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
},
{
  "Sid": "CreateEMRTaggedLaunchTemplate",
  "Effect": "Allow",
  "Action": "ec2:CreateLaunchTemplate",
  "Resource": "arn:aws:ec2:*:*:launch-template/*",
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
},
{
  "Sid": "CreateEMRTaggedInstancesAndVolumes",
  "Effect": "Allow",
  "Action": [
    "ec2:RunInstances",
    "ec2:CreateFleet"
  ],
  "Resource": [

```

```

    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:volume/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
},
{
  "Sid": "ResourcesToLaunchEC2",
  "Effect": "Allow",
  "Action": [
    "ec2:RunInstances",
    "ec2:CreateFleet",
    "ec2:CreateLaunchTemplate",
    "ec2:CreateLaunchTemplateVersion"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*:*:image/ami-*",
    "arn:aws:ec2:*:*:key-pair/*",
    "arn:aws:ec2:*:*:capacity-reservation/*",
    "arn:aws:ec2:*:*:placement-group/pg-*",
    "arn:aws:ec2:*:*:fleet/*",
    "arn:aws:ec2:*:*:dedicated-host/*",
    "arn:aws:resource-groups:*:*:group/*"
  ]
},
{
  "Sid": "ManageEMRTaggedResources",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateLaunchTemplateVersion",
    "ec2>DeleteLaunchTemplate",
    "ec2>DeleteNetworkInterface",
    "ec2:ModifyInstanceAttribute",
    "ec2:TerminateInstances"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
}

```



```

    }
  },
  {
    "Sid": "ManageTagsOnEMRTaggedResources",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags",
      "ec2:DeleteTags"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:instance/*",
      "arn:aws:ec2:*:*:volume/*",
      "arn:aws:ec2:*:*:network-interface/*",
      "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
      }
    }
  },
  {
    "Sid": "CreateNetworkInterfaceNeededForPrivateSubnet",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:network-interface/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
      }
    }
  },
  {
    "Sid": "TagOnCreateTaggedEMRResources",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:network-interface/*",

```

```

    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:volume/*",
    "arn:aws:ec2:*:*:launch-template/*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": [
        "RunInstances",
        "CreateFleet",
        "CreateLaunchTemplate",
        "CreateNetworkInterface"
      ]
    }
  },
  {
    "Sid": "TagPlacementGroups",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags",
      "ec2:DeleteTags"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:placement-group/pg-*"
    ]
  },
  {
    "Sid": "ListActionsForEC2Resources",
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeAccountAttributes",
      "ec2:DescribeCapacityReservations",
      "ec2:DescribeDhcpOptions",
      "ec2:DescribeImages",
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceTypeOfferings",
      "ec2:DescribeLaunchTemplates",
      "ec2:DescribeNetworkAcls",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribePlacementGroups",
      "ec2:DescribeRouteTables",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ec2:DescribeVolumes",

```

```

    "ec2:DescribeVolumeStatus",
    "ec2:DescribeVpcAttribute",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcs"
  ],
  "Resource": "*"
},
{
  "Sid": "CreateDefaultSecurityGroupWithEMRTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateSecurityGroup"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:security-group/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
},
{
  "Sid": "CreateDefaultSecurityGroupInVPCWithEMRTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateSecurityGroup"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:vpc/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
},
{
  "Sid": "TagOnCreateDefaultSecurityGroupWithEMRTags",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:*:*:security-group/*",

```

```

"Condition": {
  "StringEquals": {
    "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
    "ec2:CreateAction": "CreateSecurityGroup"
  }
},
{
  "Sid": "ManageSecurityGroups",
  "Effect": "Allow",
  "Action": [
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:AuthorizeSecurityGroupIngress",
    "ec2:RevokeSecurityGroupEgress",
    "ec2:RevokeSecurityGroupIngress"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
  }
},
{
  "Sid": "CreateEMRPlacementGroups",
  "Effect": "Allow",
  "Action": [
    "ec2:CreatePlacementGroup"
  ],
  "Resource": "arn:aws:ec2:*:*:placement-group/pg-*"
},
{
  "Sid": "DeletePlacementGroups",
  "Effect": "Allow",
  "Action": [
    "ec2:DeletePlacementGroup"
  ],
  "Resource": "*"
},
{
  "Sid": "AutoScaling",
  "Effect": "Allow",
  "Action": [
    "application-autoscaling:DeleteScalingPolicy",

```

```

    "application-autoscaling:DeregisterScalableTarget",
    "application-autoscaling:DescribeScalableTargets",
    "application-autoscaling:DescribeScalingPolicies",
    "application-autoscaling:PutScalingPolicy",
    "application-autoscaling:RegisterScalableTarget"
  ],
  "Resource": "*"
},
{
  "Sid": "ResourceGroupsForCapacityReservations",
  "Effect": "Allow",
  "Action": [
    "resource-groups:ListGroupResources"
  ],
  "Resource": "*"
},
{
  "Sid": "AutoScalingCloudWatch",
  "Effect": "Allow",
  "Action": [
    "cloudwatch:PutMetricAlarm",
    "cloudwatch:DeleteAlarms",
    "cloudwatch:DescribeAlarms"
  ],
  "Resource": "arn:aws:cloudwatch:*:*:alarm:*_EMR_Auto_Scaling"
},
{
  "Sid": "PassRoleForAutoScaling",
  "Effect": "Allow",
  "Action": "iam:PassRole",
  "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
  "Condition": {
    "StringLike": {
      "iam:PassedToService": "application-autoscaling.amazonaws.com*"
    }
  }
},
{
  "Sid": "PassRoleForEC2",
  "Effect": "Allow",
  "Action": "iam:PassRole",
  "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
  "Condition": {
    "StringLike": {

```

```

    "iam:PassedToService": "ec2.amazonaws.com*"
  }
},
{
  "Sid": "CreateAndModifyEmrServiceVPCEndpoint",
  "Effect": "Allow",
  "Action": [
    "ec2:ModifyVpcEndpoint",
    "ec2:CreateVpcEndpoint"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:vpc-endpoint/*",
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*",
    "arn:aws:ec2:*:*:vpc/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-
policies": "true"
    }
  }
},
{
  "Sid": "CreateEmrServiceVPCEndpoint",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateVpcEndpoint"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:vpc-endpoint/*"
  ],
  "Condition": {
    "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-
policies": "true",
      "aws:RequestTag/Name": "emr-service-vpce"
    }
  }
},
{
  "Sid": "TagEmrServiceVPCEndpoint",
  "Effect": "Allow",

```

```

        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction": "CreateVpcEndpoint",
                "aws:RequestTag/for-use-with-amazon-emr-managed-
policies": "true",
                "aws:RequestTag/Name": "emr-service-vpce"
            }
        }
    }
}

```

Ihre Servicerolle sollte die folgende Vertrauensrichtlinie verwenden.

Important

Die folgende Vertrauensrichtlinie umfasst die globalen Bedingungsschlüssel [aws:SourceArn](#) und [aws:SourceAccount](#), die die Berechtigungen einschränken, die Sie Amazon EMR auf bestimmte Ressourcen in Ihrem Konto erteilen. Auf diese Weise können Sie sich vor dem [Problem des verwirrten Stellvertreters](#) schützen.

```

{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}

```

```

    }
  }
}

```

Servicerolle für EC2 Cluster-Instances (EC2Instance-Profil)

Die Servicerolle für EC2 Cluster-Instances (auch EC2 Instance-Profil für Amazon EMR genannt) ist eine spezielle Art von Servicerolle, die jeder EC2 Instance in einem Amazon EMR-Cluster zugewiesen wird, wenn die Instance gestartet wird. Anwendungsprozesse, die auf der Hadoop-Ökosystem ausgeführt werden, übernehmen diese Rolle für Berechtigungen für die Interaktion mit anderen AWS -Services.

Weitere Informationen zu Servicerollen für EC2 Instances finden Sie im IAM-Benutzerhandbuch [unter Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden.

Important

Die Standard-Servicerolle für EC2 Cluster-Instances und die zugehörige verwaltete AWS Standardrichtlinie `AmazonElasticMapReduceforEC2Role` sind inzwischen veraltet, sodass keine neuen verwalteten Richtlinien bereitgestellt werden. AWS Sie müssen ein Instance-Profil erstellen und angeben, um die veraltete Rolle und die Standardrichtlinie zu ersetzen.

Standardrolle und verwaltete Richtlinie

- Der Standardrollenname ist `EMR_EC2_DefaultRole`.
- Die `EMR_EC2_DefaultRole` standardmäßige verwaltete Richtlinie, `AmazonElasticMapReduceforEC2Role`, nähert sich dem Ende des Supports. Anstatt eine verwaltete Standardrichtlinie für das EC2 Instance-Profil zu verwenden, wenden Sie ressourcenbasierte Richtlinien auf S3-Buckets und andere Ressourcen an, die Amazon EMR benötigt, oder verwenden Sie Ihre eigene, vom Kunden verwaltete Richtlinie mit einer IAM-Rolle als Instance-Profil. Weitere Informationen finden Sie unter [Erstellen Sie eine Servicerolle für Cluster-Instances mit Berechtigungen mit den geringsten Rechten EC2](#).

Im Folgenden werden die Inhalte von Version 3 von `AmazonElasticMapReduceforEC2Role` gezeigt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": "*",
      "Action": [
        "cloudwatch:*",
        "dynamodb:*",
        "ec2:Describe*",
        "elasticmapreduce:Describe*",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:ListClusters",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:ListSteps",
        "kinesis:CreateStream",
        "kinesis>DeleteStream",
        "kinesis:DescribeStream",
        "kinesis:GetRecords",
        "kinesis:GetShardIterator",
        "kinesis:MergeShards",
        "kinesis:PutRecord",
        "kinesis:SplitShard",
        "rds:Describe*",
        "s3:*",
        "sdb:*",
        "sns:*",
        "sqs:*",
        "glue:CreateDatabase",
        "glue:UpdateDatabase",
        "glue>DeleteDatabase",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:CreateTable",
        "glue:UpdateTable",
        "glue>DeleteTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetTableVersions",
```

```

        "glue:CreatePartition",
        "glue:BatchCreatePartition",
        "glue:UpdatePartition",
        "glue>DeletePartition",
        "glue:BatchDeletePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition",
        "glue:CreateUserDefinedFunction",
        "glue:UpdateUserDefinedFunction",
        "glue>DeleteUserDefinedFunction",
        "glue:GetUserDefinedFunction",
        "glue:GetUserDefinedFunctions"
    ]
}
]
}

```

Ihre Servicerolle sollte die folgende Vertrauensrichtlinie verwenden.

```

{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Erstellen Sie eine Servicerolle für Cluster-Instances mit Berechtigungen mit den geringsten Rechten EC2

Als bewährte Methode empfehlen wir dringend, eine Servicerolle für EC2 Cluster-Instances und eine Berechtigungsrichtlinie zu erstellen, die über die Mindestberechtigungen für andere AWS Dienste verfügt, die für Ihre Anwendung erforderlich sind.

Die standardmäßige verwaltete Richtlinie, `AmazonElasticMapReduceforEC2Role`, bietet Berechtigungen, mit denen Sie problemlos einen ersten Cluster starten können.

AmazonElasticMapReduceforEC2Roleist jedoch auf dem Weg, veraltet zu werden, und Amazon EMR wird keine AWS Ersatz-verwaltete Standardrichtlinie für die veraltete Rolle bereitstellen. Um einen ersten Cluster zu starten, müssen Sie eine vom Kunden verwaltete, ressourcenbasierte oder ID-basierte Richtlinie bereitstellen.

Die Richtlinienanweisungen unten zeigen Beispiele für die für verschiedene Features von Amazon EMR erforderlichen Berechtigungen. Wir empfehlen, diese Berechtigungen zu verwenden, um eine Berechtigungsrichtlinie zu erstellen, die den Zugriff auf nur diese Funktionen und Ressourcen beschränkt, die Ihr Cluster erfordert. Alle Beispielrichtlinien verwenden die *us-west-2* Region und die fiktive Konto-ID. AWS *123456789012* Ersetzen Sie diese je nach Bedarf für Ihren Cluster.

Weitere Informationen zum Erstellen und Angeben benutzerdefinierter Rollen finden Sie unter [Passen Sie IAM-Rollen mit Amazon EMR an](#).

 Note

Wenn Sie eine benutzerdefinierte EMR-Rolle für erstellen EC2, folgen Sie dem grundlegenden Arbeitsablauf, der automatisch ein Instanzprofil mit demselben Namen erstellt. Amazon EC2 ermöglicht es Ihnen, Instance-Profile und Rollen mit unterschiedlichen Namen zu erstellen, aber Amazon EMR unterstützt diese Konfiguration nicht und führt zu einem Fehler „Ungültiges Instance-Profil“, wenn Sie den Cluster erstellen.

Lesen und Schreiben von Daten in Amazon S3 mithilfe von EMRFS

Wenn eine Anwendung, die auf einem Amazon EMR-Cluster ausgeführt wird, Daten im *s3://mydata* Format referenziert, verwendet Amazon EMR das EC2 Instance-Profil, um die Anfrage zu stellen. Cluster lesen und schreiben Daten in der Regel auf diese Weise in Amazon S3, und Amazon EMR verwendet standardmäßig die mit der Service-Rolle verknüpften Berechtigungen für EC2 Cluster-Instances. Weitere Informationen finden Sie unter [Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3](#).

Da IAM-Rollen für EMRFS auf die mit der Servicerolle für EC2 Cluster-Instances verknüpften Berechtigungen zurückgreifen, empfehlen wir als bewährte Methode, dass Sie IAM-Rollen für EMRFS verwenden und die EMRFS- und Amazon S3 S3-Berechtigungen einschränken, die mit der Servicerolle für Cluster-Instances verknüpft sind. EC2

Die Beispielanweisung unten zeigt die Berechtigungen, die erforderlich sind, damit EMRFS Anforderungen an Amazon S3 senden kann.

- *my-data-bucket-in-s3-for-emrfs-reads-and-writes* spezifiziert den Bucket in Amazon S3, mit dem der Cluster Daten liest und schreibt, sowie alle Unterordner mit */**. Fügen Sie nur die Buckets und Ordner hinzu, die Ihre Anwendung benötigt.
- Die Richtlinienenerklärung, die dynamodb-Aktionen zulässt, ist nur erforderlich, wenn die konsistente EMRFS-Ansicht aktiviert ist. *EmrFSMetadata* gibt den Standardordner für die konsistente EMRFS-Ansicht an.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:AbortMultipartUpload",
        "s3:CreateBucket",
        "s3:DeleteObject",
        "s3:GetBucketVersioning",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:ListBucket",
        "s3:ListBucketMultipartUploads",
        "s3:ListBucketVersions",
        "s3:ListMultipartUploadParts",
        "s3:PutBucketVersioning",
        "s3:PutObject",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes",
        "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:CreateTable",
        "dynamodb:BatchGetItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:PutItem",
        "dynamodb:DescribeTable",
```

```

        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:Scan",
        "dynamodb:Query",
        "dynamodb:UpdateItem",
        "dynamodb:DeleteTable",
        "dynamodb:UpdateTable"
    ],
    "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/EmrFSMetadata"
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:PutMetricData",
        "dynamodb:ListTables",
        "s3:ListBucket"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "sqs:GetQueueUrl",
        "sqs:ReceiveMessage",
        "sqs>DeleteQueue",
        "sqs:SendMessage",
        "sqs:CreateQueue"
    ],
    "Resource": "arn:aws:sqs:us-west-2:123456789012:EMRFS-Inconsistency-*"
}
]
}

```

Archivieren von Protokolldateien in Amazon S3

Die folgende Richtlinienanweisung ermöglicht dem Amazon-EMR-Cluster die Archivierung von Protokolldateien in dem angegebenen Amazon-S3-Speicherort. Im folgenden Beispiel wurde bei der Erstellung des Clusters der Speicherort Protokollordner S3 in der Konsole, mithilfe der `--log-uri` Option von oder mithilfe des `LogUri` Parameters im `RunJobFlow` Befehl angegeben. *s3://MyLoggingBucket/MyEMRClusterLogs* AWS CLI Weitere Informationen finden Sie unter [Archivieren von Protokolldateien in Amazon S3](#).

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::MyLoggingBucket/MyEMRClusterLogs/*"
  }
]
}

```

Verwenden des AWS Glue-Datenkatalogs

Die folgende Richtlinienerklärung erlaubt Aktionen, die erforderlich sind, wenn Sie den AWS Glue-Datenkatalog als Metastore für Anwendungen verwenden. Weitere Informationen finden Sie unter [Verwenden des AWS Glue-Datenkatalogs als Metastore für Spark SQL](#), [Verwenden des AWS Glue-Datenkatalogs als Metastore für Hive](#) und [Verwenden von Presto mit dem AWS Glue-Datenkatalog](#) im Amazon EMR-Versionshandbuch.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase",
        "glue:UpdateDatabase",
        "glue>DeleteDatabase",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:CreateTable",
        "glue:UpdateTable",
        "glue>DeleteTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetTableVersions",
        "glue:CreatePartition",
        "glue:BatchCreatePartition",
        "glue:UpdatePartition",
        "glue>DeletePartition",
        "glue:BatchDeletePartition",
        "glue:GetPartition",
        "glue:GetPartitions",

```

```

        "glue:BatchGetPartition",
        "glue:CreateUserDefinedFunction",
        "glue:UpdateUserDefinedFunction",
        "glue>DeleteUserDefinedFunction",
        "glue:GetUserDefinedFunction",
        "glue:GetUserDefinedFunctions"
    ],
    "Resource": "*",
}
]
}

```

Servicerolle für Auto Scaling in Amazon EMR (Auto Scaling-Rolle)

Die Auto-Scaling-Rolle für Amazon EMR führt eine ähnliche Funktion aus wie die Servicerolle, ermöglicht aber zusätzliche Aktionen für dynamisch skalierte Umgebungen.

- Der Standardrollenname ist `EMR_AutoScaling_DefaultRole`.
- Die an `EMR_AutoScaling_DefaultRole` angefügte standardmäßige verwaltete Richtlinie ist `AmazonElasticMapReduceforAutoScalingRole`.

Der Inhalt von Version 1 `AmazonElasticMapReduceforAutoScalingRole` wird unten angezeigt.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ModifyInstanceGroups"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}

```

Ihre Servicerolle sollte die folgende Vertrauensrichtlinie verwenden.

⚠ Important

Die folgende Vertrauensrichtlinie umfasst die globalen Bedingungsschlüssel [aws:SourceArn](#) und [aws:SourceAccount](#), die die Berechtigungen einschränken, die Sie Amazon EMR auf bestimmte Ressourcen in Ihrem Konto erteilen. Auf diese Weise können Sie sich vor dem [Problem des verwirrten Stellvertreters](#) schützen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "application-autoscaling.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:application-
autoscaling:<region>:<account-id>:scalable-target/*"
        }
      }
    }
  ]
}
```

Servicerolle für EMR Notebooks

Jedes EMR-Notizbuch benötigt Berechtigungen, um auf andere AWS Ressourcen zuzugreifen und Aktionen auszuführen. Die mit dieser Servicerolle verknüpften IAM-Richtlinien gewähren dem Notebook Berechtigungen für die Zusammenarbeit mit anderen Diensten. AWS Wenn Sie mit dem ein Notebook erstellen AWS Management Console, geben Sie eine AWS Servicerolle an. Sie können die Standardrolle, `EMR_Notebooks_DefaultRole`, verwenden oder eine Rolle angeben, die Sie erstellen. Wenn ein Notebook nicht vorher erstellt wurde, haben Sie die Möglichkeit, die Standardrolle zu erstellen.

- Der Standardrollenname ist `EMR_Notebooks_DefaultRole`.
- Die standardmäßig angehängten verwalteten Richtlinien zu `EMR_Notebooks_DefaultRole` sind `AmazonElasticMapReduceEditorsRole` und `S3FullAccessPolicy`.

Ihre Servicerolle sollte die folgende Vertrauensrichtlinie verwenden.

Important

Die folgende Vertrauensrichtlinie umfasst die globalen Bedingungsschlüssel [aws:SourceArn](#) und [aws:SourceAccount](#), die die Berechtigungen einschränken, die Sie Amazon EMR auf bestimmte Ressourcen in Ihrem Konto erteilen. Auf diese Weise können Sie sich vor dem [Problem des verwirrten Stellvertreters](#) schützen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
        }
      }
    }
  ]
}
```

Der Inhalt von Version 1 von `AmazonElasticMapReduceEditorsRole` lautet wie folgt.

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

```

    {
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:CreateNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2>DeleteNetworkInterface",
        "ec2>DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DescribeTags",
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": [
            "aws:elasticmapreduce:editor-id",
            "aws:elasticmapreduce:job-flow-id"
          ]
        }
      }
    }
  ]
}

```

Im Folgenden sehen Sie den Inhalt von `S3FullAccessPolicy`. `S3FullAccessPolicy` Dadurch kann Ihre Servicerolle für EMR Notebooks alle Amazon S3-Aktionen an Objekten in Ihrem AWS-

Konto ausführen. Wenn Sie eine benutzerdefinierte Servicerolle für EMR Notebooks erstellen, müssen Sie Ihrer Servicerolle Amazon-S3-Berechtigungen erteilen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

Sie können den Lese- und Schreibzugriff für Ihre Servicerolle auf den Amazon-S3-Standort beschränken, an dem Sie Ihre Notebookdateien speichern möchten. Verwenden Sie die folgenden Mindestberechtigungen an Amazon S3.

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

Wenn Ihr Amazon-S3-Bucket verschlüsselt ist, müssen Sie die folgenden Berechtigungen für AWS Key Management Service angeben.

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

Wenn Sie Git-Repositorys mit Ihrem Notebook verknüpfen und ein Geheimnis für das Repository erstellen müssen, müssen Sie die Berechtigung `secretsmanager:GetSecretValue` in der IAM-Richtlinie hinzufügen, die der Servicerolle für Amazon EMR Notebooks zugewiesen ist. Eine Beispielryrichtlinie wird nachfolgend gezeigt:

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

```

    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

Berechtigungen für die Servicerolle von EMR Notebooks

In dieser Tabelle sind die Aktionen aufgeführt, die EMR Notebooks mithilfe der Servicerolle durchführt, zusammen mit den Berechtigungen, die für jede Aktion erforderlich sind.

Aktion	Berechtigungen
Richten Sie einen sicheren Netzwerkkanal zwischen einem Notebook und einem Amazon-EMR-Cluster ein und führen Sie die erforderlichen Bereinigungsaktionen durch.	<pre> "ec2:CreateNetworkInterface", "ec2:CreateNetworkInterfacePermission", "ec2:DeleteNetworkInterface", "ec2:DeleteNetworkInterfacePermission", "ec2:DescribeNetworkInterfaces", "ec2:ModifyNetworkInterfaceAttribute", "ec2:AuthorizeSecurityGroupEgress", "ec2:AuthorizeSecurityGroupIngress", "ec2:CreateSecurityGroup", "ec2:DescribeSecurityGroups", "ec2:RevokeSecurityGroupEgress", "ec2:DescribeTags", "ec2:DescribeInstances", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "elasticmapreduce:ListInstances", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListSteps"</pre>
Verwenden Sie die in gespeicherten Git-Anmeldeinformationen AWS Secrets Manager, um Git-Repositorys mit einem Notizbuch zu verknüpfen.	<pre> "secretsmanager:GetSecretValue"</pre>

Aktion	Berechtigungen
Wenden Sie AWS Tags auf die Netzwerkschnittstelle und die Standardsicherheitsgruppen an, die EMR Notebooks bei der Einrichtung des sicheren Netzwerkkanals erstellt. Weitere Informationen finden Sie unter Markieren von AWS -Ressourcen .	<pre>"ec2:CreateTags"</pre>
Greifen Sie auf Notebook-Dateien und Metadaten zu oder laden Sie sie in Amazon S3 hoch.	<pre>"s3:PutObject", "s3:GetObject", "s3:GetEncryptionConfiguration", "s3:ListBucket", "s3:DeleteObject"</pre> Wenn Sie einen verschlüsselten Amazon-S3-Bucket verwenden, sind die folgenden Berechtigungen erforderlich. <pre>"kms:Decrypt", "kms:GenerateDataKey", "kms:ReEncryptFrom", "kms:ReEncryptTo", "kms:DescribeKey"</pre>

Aktualisierungen der AWS verwalteten Richtlinien von EMR Notebooks

Hier finden Sie Details zu Aktualisierungen der AWS verwalteten Richtlinien für EMR Notebooks seit dem 1. März 2021.

Änderung	Beschreibung	Datum
AmazonElasticMapReduceEditorsRole - Added permissions	EMR Notebooks fügt ec2:describeVPCs - und elastmicmapreduce:	8. Februar 2023

Änderung	Beschreibung	Datum
	ListSteps -Berechtigungen für AmazonElasticMapReduceRole hinzu.	
EMR Notebooks hat damit begonnen, Änderungen zu verfolgen	EMR Notebooks begann, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	8. Februar 2023

Verwenden von serviceverknüpften Rollen für Amazon EMR

Amazon EMR verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte](#) Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EMR verknüpft ist. Servicebezogene Rollen sind von Amazon EMR vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Themen

- [Verwenden von serviceverknüpften Rollen für Amazon EMR zur Bereinigung](#)
- [Verwenden von serviceverknüpften Rollen mit Amazon EMR für die Write-Ahead-Protokollierung](#)

Informationen zu anderen Diensten, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS Dienste, die mit IAM funktionieren](#). Suchen Sie in der Spalte Serviceverknüpfte Rollen nach den Diensten, für die Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Verwenden von serviceverknüpften Rollen für Amazon EMR zur Bereinigung

Amazon EMR verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte](#) Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EMR verknüpft ist. Servicebezogene Rollen sind von Amazon EMR vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Serviceverknüpfte Rollen arbeiten mit der Amazon EMR-Servicerolle und dem EC2 Amazon-Instanzprofil für Amazon EMR zusammen. Weitere Informationen über die Service-Rolle und

das Instance-Profil finden Sie unter [Konfigurieren Sie IAM-Servicerollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen](#).

Eine serviceverknüpfte Rolle erleichtert die Einrichtung von Amazon EMR, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EMR definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon EMR seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können diese serviceverknüpfte Rolle für Amazon EMR erst löschen, nachdem Sie alle zugehörigen Ressourcen gelöscht und alle EMR-Cluster im Konto beendet haben. Dadurch werden Ihre Amazon EMR-Ressourcen geschützt, sodass Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Verwenden von Rollen, die mit Services verknüpft sind, für die Bereinigung

Amazon EMR verwendet die servicebasierte `AWSServiceRoleForEMRCleanupRole`, um Amazon EMR die Erlaubnis zu erteilen, EC2 Amazon-Ressourcen in Ihrem Namen zu beenden und zu löschen, falls die mit dem Amazon EMR-Service verknüpfte Rolle diese Fähigkeit verliert. Amazon EMR erstellt die serviceverknüpfte Rolle automatisch während der Clustererstellung, sofern sie noch nicht vorhanden ist.

Die `AWSServiceRoleForEMRCleanup` serviceverknüpfte Rolle vertraut darauf, dass die folgenden Services die Rolle übernehmen:

- `elasticmapreduce.amazonaws.com`

Die Richtlinie für `AWSServiceRoleForEMRCleanup` servicebezogene Rollenberechtigungen ermöglicht es Amazon EMR, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `DescribeInstances` für `ec2`
- Aktion: `DescribeSpotInstanceRequests` für `ec2`
- Aktion: `ModifyInstanceAttribute` für `ec2`
- Aktion: `TerminateInstances` für `ec2`
- Aktion: `CancelSpotInstanceRequests` für `ec2`
- Aktion: `DeleteNetworkInterface` für `ec2`
- Aktion: `DescribeInstanceAttribute` für `ec2`

- Aktion: DescribeVolumeStatus für ec2
- Aktion: DescribeVolumes für ec2
- Aktion: DetachVolume für ec2
- Aktion: DeleteVolume für ec2

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann.

Erstellen einer serviceverknüpften Rolle für Amazon EMR

Sie müssen die Rolle nicht manuell erstellen. AWSService RoleFor EMRCleanup Wenn Sie einen Cluster starten, entweder zum ersten Mal oder wenn die AWSService RoleFor EMRCleanup serviceverknüpfte Rolle nicht vorhanden ist, erstellt Amazon EMR die AWSService RoleFor EMRCleanup serviceverknüpfte Rolle für Sie. Sie müssen über die erforderlichen Berechtigungen verfügen, um eine serviceverknüpfte Rolle zu erstellen. Sie finden eine Beispiel-Anweisung, die diese Funktion zur Berechtigungsrichtlinie einer IAM-Entität (z. B. Benutzer, Gruppe oder Rolle) hinzufügt, unter [Verwenden von serviceverknüpften Rollen für Amazon EMR zur Bereinigung](#).

Important

Wenn Sie Amazon EMR vor dem 24. Oktober 2017 verwendet haben, als serviceverknüpfte Rollen nicht unterstützt wurden, hat Amazon EMR die AWSService RoleFor EMRCleanup serviceverknüpfte Rolle in Ihrem Konto erstellt. Weitere Informationen finden Sie unter [In meinem IAM-Konto wird eine neue Rolle angezeigt](#).

Bearbeiten einer serviceverknüpften Rolle für Amazon EMR

Amazon EMR erlaubt es Ihnen nicht, die AWSService RoleFor EMRCleanup serviceverknüpfte Rolle zu bearbeiten. Nachdem Sie eine serviceverknüpfte Rolle erstellt haben, können Sie den Namen der serviceverknüpften Rolle nicht mehr ändern, da verschiedene Entitäten möglicherweise auf die serviceverknüpfte Rolle verweisen. Sie können die Beschreibung der dienstbezogenen Rolle jedoch mithilfe von IAM bearbeiten.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus.
2. Wählen Sie den Namen der zu ändernden Rolle.
3. Wählen Sie neben Rollenbeschreibung rechts Bearbeiten aus.
4. Geben Sie eine neue Beschreibung im Dialogfeld ein und wählen Sie Save changes (Änderungen speichern).

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Befehle aus dem verwenden, AWS Command Line Interface um die Beschreibung einer serviceverknüpften Rolle zu bearbeiten.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (CLI)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die folgenden Befehle:

```
$ aws iam get-role --role-name role-name
```

Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den CLI-Befehlen zu beziehen. Wenn eine Rolle zum Beispiel folgenden ARN hat: `arn:aws:iam::123456789012:role/myrole`, verweisen Sie auf die Rolle als **myrole**.

2. Um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren, verwenden Sie einen der folgenden Befehle:

```
$ aws iam update-role-description --role-name role-name --description description
```

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (API)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie den folgenden Befehl:

IAM-API: [GetRole](#)

2. Um die Beschreibung einer Rolle zu aktualisieren, verwenden Sie den folgenden Befehl:

IAM-API: [UpdateRoleDescription](#)

Löschen einer serviceverknüpften Rolle für Amazon EMR

Wenn Sie eine Funktion oder einen Dienst, für den eine dienstverknüpfte Rolle erforderlich ist, nicht mehr verwenden müssen, empfehlen wir, diese dienstverknüpfte Rolle zu löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie löschen können.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie IAM verwenden können, um eine dienstverknüpfte Rolle zu löschen, müssen Sie zunächst sicherstellen, dass die dienstverknüpfte Rolle keine aktiven Sitzungen hat, und alle Ressourcen entfernen, die von der dienstbezogenen Rolle verwendet werden.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen. Wählen Sie den Namen (nicht das Kontrollkästchen) der serviceverknüpften Rolle aus. AWSService RoleFor EMRCleanup
3. Wählen Sie auf der Übersichtsseite für die ausgewählte dienstverknüpfte Rolle Access Advisor aus.
4. Überprüfen Sie auf der Registerkarte Access Advisor (Advisor aufrufen) die jüngsten Aktivitäten für die serviceverknüpfte Rolle.

Note

Wenn Sie sich nicht sicher sind, ob Amazon EMR die AWSService RoleFor EMRCleanup serviceverknüpfte Rolle verwendet, können Sie versuchen, die serviceverknüpfte Rolle zu löschen. Wenn der Service die serviceverknüpfte Rolle verwendet, schlägt das Löschen fehl und Sie können die Regionen anzeigen, in denen die serviceverknüpfte Rolle verwendet wird. Wenn die dienstverknüpfte Rolle verwendet wird, müssen Sie

warten, bis die Sitzung beendet ist, bevor Sie die dienstverknüpfte Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Um Amazon EMR-Ressourcen zu entfernen, die verwendet werden von AWSService RoleFor EMRCleanup

- Beenden Sie alle Cluster in Ihrem Konto. Weitere Informationen finden Sie unter [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#).

Löschen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen. Aktivieren Sie das Kontrollkästchen neben AWSService RoleForEMRCleanup, nicht den Namen oder die Zeile selbst.
3. Wählen Sie für Role actions oben auf der Seite Delete role aus.
4. Überprüfen Sie im Bestätigungsdialogfeld die Daten, auf die der Dienst zuletzt zugegriffen hat. Aus diesen Daten geht hervor, wann jede der ausgewählten Rollen zuletzt auf einen AWS Dienst zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wählen Sie Yes, Delete, um fortzufahren.
5. Sehen Sie sich die Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da das Löschen der dienstverknüpften IAM-Rolle asynchron erfolgt, kann die Löschaufgabe erfolgreich sein oder fehlschlagen, nachdem Sie die dienstverknüpfte Rolle zum Löschen eingereicht haben. Wenn der Vorgang fehlschlägt, können Sie in den Benachrichtigungen View details oder View Resources auswählen, um zu erfahren, warum die Löschung fehlgeschlagen ist. Wenn das Löschen fehlschlägt, weil der Service Ressourcen enthält, die von der Rolle verwendet werden, enthält die Angabe des Fehlergrundes eine Liste der Ressourcen.

Löschen einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Befehle von verwenden, um eine dienstverknüpfte Rolle AWS Command Line Interface zu löschen. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie

verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschanforderung übermitteln. Wenn diese Bedingungen nicht erfüllt sind, kann diese Anforderung verweigert werden.

So löschen Sie eine serviceverknüpfte Rolle (CLI)

1. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen. Geben Sie den folgenden Befehl ein, um eine Löschanforderung für eine serviceverknüpfte Rolle zu übermitteln:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForEMRCleanup
```

2. Geben Sie den folgenden Befehl ein, um den Status der Löschaufgabe zu überprüfen:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Der Status der Löschaufgabe kann NOT_STARTED, IN_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Löschen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API zum Löschen einer serviceverknüpften Rolle verwenden. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschanforderung übermitteln. Wenn diese Bedingungen nicht erfüllt sind, kann diese Anforderung verweigert werden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Rufen Sie an, um eine Löschanfrage für eine dienstverknüpfte Rolle einzureichen. [DeleteServiceLinkedRole](#) Geben Sie in der Anfrage den AWSService RoleFor EMRCleanup Rollennamen an.

Sie benötigen die `DeletionTaskId` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

2. Um den Status der Löschung zu überprüfen, rufen Sie [GetServiceLinkedRoleDeletionStatus](#) auf. Geben Sie in der Anforderung die `DeletionTaskId` an.

Der Status der Löschaufgabe kann NOT_STARTED, IN_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Unterstützte Regionen für AWSService RoleFor EMRCleanup

Amazon EMR unterstützt die Verwendung der AWSService RoleFor EMRCleanup serviceverknüpften Rolle in den folgenden Regionen.

Name der Region	Regions-ID	Amazon EMR Support
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Irland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja

Name der Region	Regions-ID	Amazon EMR Support
Südamerika (São Paulo)	sa-east-1	Ja

Verwenden von serviceverknüpften Rollen mit Amazon EMR für die Write-Ahead-Protokollierung

Amazon EMR verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte](#) Rollen. Eine serviceverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit Amazon EMR verknüpft ist. Servicebezogene Rollen sind von Amazon EMR vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

Serviceverknüpfte Rollen arbeiten mit der Amazon EMR-Servicerolle und dem EC2 Amazon-Instanzprofil für Amazon EMR zusammen. Weitere Informationen über die Service-Rolle und das Instance-Profil finden Sie unter [Konfigurieren Sie IAM-Servicerollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen](#).

Eine serviceverknüpfte Rolle erleichtert die Einrichtung von Amazon EMR, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon EMR definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon EMR seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können diese serviceverknüpfte Rolle für Amazon EMR erst löschen, nachdem Sie die zugehörigen Ressourcen gelöscht und alle EMR-Cluster im Konto beendet haben. Dadurch werden Ihre Amazon EMR-Ressourcen geschützt, sodass Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Servicebezogene Rollenberechtigungen für Write-Ahead Logging (WAL)

Amazon EMR verwendet die serviceverknüpfte Rolle `AWSServiceRoleForEMRWAL`, um einen Cluster-Status abzurufen.

Die serviceverknüpfte `AWSService RoleFor EMRWAL`-Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `emrwal.amazonaws.com`

Die [EMRDescribeClusterPolicyForEMRWAL](#) Berechtigungsrichtlinie für die serviceverknüpfte Rolle ermöglicht es Amazon EMR, die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `DescribeCluster` für *

Sie müssen Berechtigungen konfigurieren, damit eine IAM-Entität (in diesem Fall Amazon EMR WAL) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Fügen Sie der Berechtigungsrichtlinie für Ihr Instance-Profil nach Bedarf die folgenden Anweisungen hinzu:

CreateServiceLinkedRole

Um einer IAM-Entität zu ermöglichen, die serviceverknüpfte `AWSServiceRoleForEMRWAL`-Rolle zu erstellen

Fügen Sie die folgende Anweisung der Berechtigungsrichtlinie für die IAM-Entität hinzu, die die serviceverknüpfte Rolle erstellen soll:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole",
    "iam:PutRolePolicy"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/emrwal.amazonaws.com*/AWSServiceRoleForEMRWAL*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": [
        "emrwal.amazonaws.com",
        "elasticmapreduce.amazonaws.com.cn"
      ]
    }
  }
}
```

UpdateRoleDescription

Um es einer IAM-Entität zu ermöglichen, die Beschreibung der serviceverknüpften EMRWAL-Rolle zu bearbeiten `AWSServiceRoleFor`

Fügen Sie die folgende Anweisung der Berechtigungsrichtlinie für die IAM-Entität hinzu, die die Beschreibung einer serviceverknüpften Rolle bearbeiten soll:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:UpdateRoleDescription"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/emrwal.amazonaws.com*/
AWSServiceRoleForEMRWAL*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": [
        "emrwal.amazonaws.com",
        "elasticmapreduce.amazonaws.com.cn"
      ]
    }
  }
}
```

DeleteServiceLinkedRole

Um einer IAM-Entität zu ermöglichen, die mit dem EMRWAL-Service verknüpfte Rolle zu löschen
AWSService RoleFor

Fügen Sie die folgende Anweisung der Berechtigungsrichtlinie für die IAM-Entität hinzu, die eine serviceverknüpfte Rolle löschen soll:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
  "Condition": {
    "StringLike": {
      "iam:AWSServiceName": [
        "emrwal.amazonaws.com",
        "elasticmapreduce.amazonaws.com.cn"
      ]
    }
  }
}
```



```
}  
}  
}
```

Erstellen einer serviceverknüpften Rolle für Amazon EMR

Sie müssen die EMRWAL-Rolle nicht manuell erstellen. AWSService RoleFor Amazon EMR erstellt diese serviceverknüpfte Rolle automatisch, wenn Sie einen WAL-Workspace mit der EMRWAL-CLI oder von dort aus erstellen AWS CloudFormation, oder erstellt die serviceverknüpfte Rolle, wenn Sie einen Workspace für Amazon EMR WAL konfigurieren und die serviceverknüpfte Rolle HBase noch nicht existiert. Sie müssen über die erforderlichen Berechtigungen verfügen, um eine serviceverknüpfte Rolle zu erstellen. Anweisungen, mit denen diese Funktion zur Berechtigungsrichtlinie einer IAM-Entität (z. B. eines Benutzers, einer Gruppe oder einer Rolle) hinzugefügt wird, finden Sie beispielsweise im vorherigen Abschnitt. [Servicebezogene Rollenberechtigungen für Write-Ahead Logging \(WAL\)](#)

Bearbeiten einer serviceverknüpften Rolle für Amazon EMR

Amazon EMR erlaubt Ihnen nicht, die mit dem AWSService RoleFor EMRWAL-Service verknüpfte Rolle zu bearbeiten. Nachdem Sie eine serviceverknüpfte Rolle erstellt haben, können Sie den Namen der serviceverknüpften Rolle nicht mehr ändern, da verschiedene Entitäten möglicherweise auf die serviceverknüpfte Rolle verweisen. Sie können die Beschreibung der dienstbezogenen Rolle jedoch mithilfe von IAM bearbeiten.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus.
2. Wählen Sie den Namen der zu ändernden Rolle.
3. Wählen Sie neben Rollenbeschreibung rechts Bearbeiten aus.
4. Geben Sie eine neue Beschreibung im Dialogfeld ein und wählen Sie Save changes (Änderungen speichern).

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Befehle aus dem verwenden, AWS Command Line Interface um die Beschreibung einer serviceverknüpften Rolle zu bearbeiten.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (CLI)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die folgenden Befehle:

```
$ aws iam get-role --role-name role-name
```

Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den CLI-Befehlen zu beziehen. Wenn eine Rolle zum Beispiel folgenden ARN hat: `arn:aws:iam::123456789012:role/myrole`, verweisen Sie auf die Rolle als **myrole**.

2. Um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren, verwenden Sie einen der folgenden Befehle:

```
$ aws iam update-role-description --role-name role-name --description description
```

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (API)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie den folgenden Befehl:

IAM-API: [GetRole](#)

2. Um die Beschreibung einer Rolle zu aktualisieren, verwenden Sie den folgenden Befehl:

IAM-API: [UpdateRoleDescription](#)

Löschen einer serviceverknüpften Rolle für Amazon EMR

Wenn Sie eine Funktion oder einen Dienst, für den eine dienstverknüpfte Rolle erforderlich ist, nicht mehr verwenden müssen, empfehlen wir, diese dienstverknüpfte Rolle zu löschen. Auf diese Weise

haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie löschen können.

 Note

Der Write-Ahead-Protokollierungsvorgang ist nicht betroffen, wenn Sie die AWSService RoleFor EMRWAL-Rolle löschen, aber Amazon EMR löscht die erstellten Protokolle nicht automatisch, sobald Ihr EMR-Cluster beendet wird. Daher müssen Sie die Amazon EMR WAL-Protokolle manuell löschen, wenn Sie die serviceverknüpfte Rolle löschen.

Bereinigen einer serviceverknüpften Rolle

Bevor Sie mit IAM eine serviceverknüpfte Rolle löschen können, müssen Sie sich zunächst vergewissern, dass die Rolle über keine aktiven Sitzungen verfügt, und alle Ressourcen entfernen, die von der Rolle verwendet werden.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen. Wählen Sie den Namen (nicht das Kontrollkästchen) der AWSService RoleFor EMRWAL-Rolle aus.
3. Wählen Sie auf der Seite Summary (Übersicht) für die ausgewählte Rolle die Option Access Advisor (Advisor aufrufen) aus.
4. Überprüfen Sie auf der Registerkarte Access Advisor (Advisor aufrufen) die jüngsten Aktivitäten für die serviceverknüpfte Rolle.

 Note

Wenn Sie sich nicht sicher sind, ob Amazon EMR die AWSService RoleFor EMRWAL-Rolle verwendet, können Sie versuchen, die serviceverknüpfte Rolle zu löschen. Wenn der Service die Rolle verwendet, schlägt das Löschen fehl und Sie können die Regionen anzeigen, in denen die mit dem Service verknüpfte Rolle verwendet wird. Wenn die dienstverknüpfte Rolle verwendet wird, müssen Sie warten, bis die Sitzung beendet ist, bevor Sie die dienstverknüpfte Rolle löschen können. Die Sitzung für eine serviceverknüpfte Rolle können Sie nicht widerrufen.

Um Amazon EMR-Ressourcen zu entfernen, die von der AWSService RoleFor EMRWAL verwendet werden

- Beenden Sie alle Cluster in Ihrem Konto. Weitere Informationen finden Sie unter [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#).

Löschen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen. Aktivieren Sie das Kontrollkästchen neben AWSService RoleFor EMRWAL, nicht den Namen oder die Zeile selbst.
3. Wählen Sie für Role actions oben auf der Seite Delete role aus.
4. Überprüfen Sie im Bestätigungsdialogfeld die Daten, auf die der Dienst zuletzt zugegriffen hat. Aus diesen Daten geht hervor, wann jede der ausgewählten Rollen zuletzt auf einen AWS Dienst zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wählen Sie Yes, Delete, um fortzufahren.
5. Sehen Sie sich die Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da die Löschung der serviceverknüpften IAM-Rolle asynchron erfolgt, kann die Löschung nach dem Übermitteln der Rolle für die Löschung erfolgreich sein oder fehlschlagen. Wenn der Vorgang fehlschlägt, können Sie in den Benachrichtigungen View details oder View Resources auswählen, um zu erfahren, warum die Löschung fehlgeschlagen ist. Wenn das Löschen fehlschlägt, weil der Service Ressourcen enthält, die von der Rolle verwendet werden, enthält die Angabe des Fehlergrundes eine Liste der Ressourcen.

Löschen einer serviceverknüpften Rolle (IAM-CLI)

Sie können IAM-Befehle von verwenden, AWS Command Line Interface um eine dienstverknüpfte Rolle zu löschen. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschanforderung übermitteln. Wenn diese Bedingungen nicht erfüllt sind, kann diese Anforderung verweigert werden.

So löschen Sie eine serviceverknüpfte Rolle (CLI)

1. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen. Geben Sie den folgenden Befehl ein, um eine Löschanforderung für eine serviceverknüpfte Rolle zu übermitteln:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForEMRWAL
```

2. Geben Sie den folgenden Befehl ein, um den Status der Löschaufgabe zu überprüfen:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Der Status der Löschaufgabe kann NOT_STARTED, IN_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Löschen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API zum Löschen einer serviceverknüpften Rolle verwenden. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschanforderung übermitteln. Wenn diese Bedingungen nicht erfüllt sind, kann diese Anforderung verweigert werden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Rufen Sie an, um eine Löschanfrage für eine dienstverknüpfte Rolle einzureichen. [DeleteServiceLinkedRole](#) Geben Sie in der Anfrage den AWSService RoleFor EMRWAL-Rollenamen an.

Sie benötigen die `DeletionTaskId` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

2. Um den Status der Löschung zu überprüfen, rufen Sie [GetServiceLinkedRoleDeletionStatus](#) auf. Geben Sie in der Anforderung die `DeletionTaskId` an.

Der Status der Löschaufgabe kann NOT_STARTED, IN_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Unterstützte Regionen für EMRWAL AWSService RoleFor

Amazon EMR unterstützt die Verwendung der serviceverknüpften AWSService RoleFor EMRWAL-Rolle in den folgenden Regionen.

Name der Region	Regions-ID	Amazon EMR Support
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja

Passen Sie IAM-Rollen mit Amazon EMR an

Sie können die IAM-Servicerollen und -Berechtigungen anpassen, um Rechte entsprechend Ihren Sicherheitsanforderungen zu beschränken. Zum Anpassen von Berechtigungen empfehlen wir, dass Sie neue Rollen und Richtlinien erstellen. Beginnen Sie mit den Berechtigungen in den verwalteten Richtlinien für die Standardrollen (beispielsweise `AmazonElasticMapReduceforEC2Role` und `AmazonElasticMapReduceRole`). Kopieren Sie anschließend die Inhalte in die neuen Richtlinienanweisungen, modifizieren Sie die Berechtigungen entsprechend und fügen Sie die geänderten Richtlinien zu den von Ihnen erstellten Rollen hinzu. Sie müssen über die entsprechenden IAM-Berechtigungen für die Arbeit mit Rollen und Richtlinien verfügen. Weitere Informationen finden Sie unter [Benutzern und Gruppen gestatten, Rollen zu erstellen und zu ändern](#).

Wenn Sie eine benutzerdefinierte EMR-Rolle für erstellen EC2, folgen Sie dem grundlegenden Arbeitsablauf, der automatisch ein Instanzprofil mit demselben Namen erstellt. Amazon EC2 ermöglicht es Ihnen, Instance-Profile und Rollen mit unterschiedlichen Namen zu erstellen, aber Amazon EMR unterstützt diese Konfiguration nicht und führt zu einem Fehler „Ungültiges Instance-Profil“, wenn Sie den Cluster erstellen.

 Important

Eingebundene Richtlinien werden nicht automatisch aktualisiert, wenn sich Serviceanforderungen ändern. Beachten Sie beim Erstellen und Anhängen von Inline-Richtlinien, dass es zu Serviceaktualisierungen kommen kann, die plötzlich zu Berechtigungsfehlern führen. Weitere Informationen hierzu finden Sie unter [Verwaltete Richtlinien und eingebundene Richtlinien](#) im IAM-Benutzerhandbuch und [Angabe benutzerdefinierter IAM-Rollen beim Erstellen eines Clusters](#).

Weitere Informationen über die Arbeit mit IAM-Rollen finden Sie in den folgenden Themen im IAM-Benutzerhandbuch:

- [Eine Rolle erstellen, um Berechtigungen an einen Service zu delegieren AWS](#)
- [Ändern einer Rolle](#)
- [Löschen einer Rolle](#)

Angabe benutzerdefinierter IAM-Rollen beim Erstellen eines Clusters

Sie geben die Servicerolle für Amazon EMR und die Rolle für das EC2 Amazon-Instance-Profil an, wenn Sie einen Cluster erstellen. Der Benutzer, der Cluster erstellt, benötigt Berechtigungen zum Abrufen und Zuweisen von Rollen zu Amazon EMR und EC2 Instances. Andernfalls tritt der EC2 Fehler „Ein Konto ist nicht zum Anrufen autorisiert“ auf. Weitere Informationen finden Sie unter [Benutzern und Gruppen gestatten, Rollen zu erstellen und zu ändern](#).

Mit der Konsole benutzerdefinierte Rollen angeben

Wenn Sie einen Cluster erstellen, können Sie mithilfe der erweiterten Optionen eine benutzerdefinierte Servicerolle für Amazon EMR, eine benutzerdefinierte Rolle für das EC2 Instance-Profil und eine benutzerdefinierte Auto Scaling Scaling-Rolle angeben. Wenn Sie Quick Options verwenden, werden die Standard-Servicerolle und die Standardrolle für das EC2 Instance-

Profil angegeben. Weitere Informationen finden Sie unter [Von Amazon EMR verwendete IAM-Servicerollen](#).

Console

Um benutzerdefinierte IAM-Rollen mit der Konsole anzugeben

Wenn Sie mit der Konsole einen Cluster erstellen, müssen Sie eine benutzerdefinierte Servicerolle für Amazon EMR und eine benutzerdefinierte Rolle für das EC2 Instance-Profil angeben. Weitere Informationen finden Sie unter [Von Amazon EMR verwendete IAM-Servicerollen](#).

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Suchen Sie unter Sicherheitskonfiguration und Berechtigungen die Felder IAM-Rolle für Instance-Profil und Servicerolle für Amazon EMR. Wählen Sie für jeden Rollentyp eine Rolle aus der Liste aus. Nur Rollen innerhalb Ihres Kontos, die die entsprechende Vertrauensstellungen für diesen Rollentyp besitzen, sind aufgeführt.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

Verwenden Sie die AWS CLI , um benutzerdefinierte Rollen anzugeben

Sie können eine Servicerolle für Amazon EMR und eine Servicerolle für EC2 Cluster-Instances explizit mithilfe von Optionen mit dem `create-cluster` Befehl von angeben. AWS CLI Verwenden Sie die Option `--service-role`, um die Servicerolle anzugeben. Verwenden Sie das `InstanceProfile` Argument der `--ec2-attributes` Option, um die Rolle für das EC2 Instance-Profil anzugeben.

Die Auto Scaling-Rolle wird einer separaten Option angegeben, `--auto-scaling-role`. Weitere Informationen finden Sie unter [Verwenden der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instanzgruppen in Amazon EMR](#).

Um benutzerdefinierte IAM-Rollen anzugeben, verwenden Sie AWS CLI

- Der folgende Befehl gibt die benutzerdefinierte Servicerolle und eine benutzerdefinierte Rolle für das EC2 Instanzprofil an *MyCustomServiceRoleForClusterEC2Instances*, wenn ein

Cluster gestartet wird. *MyCustomServiceRoleForEMR* Dieses Beispiel verwendet die Amazon-EMR-Standardrolle.

 Note

Linux-Zeilenfortsetzungszeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.9.0 \  
--applications Name=Hive Name=Pig --service-role MyCustomServiceRoleForEMR \  
--ec2-attributes InstanceProfile=MyCustomServiceRoleForClusterEC2Instances,\  
KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

Sie können diese Optionen verwenden, um Standardrollen explizit anzugeben, statt die Option `--use-default-roles` zu verwenden. Die `--use-default-roles` Option gibt die Servic Rolle und die Rolle für das EC2 Instanzprofil an, die in der config Datei für definiert sind AWS CLI.

Das folgende Beispiel zeigt den Inhalt einer config Datei für AWS CLI die spezifizierten benutzerdefinierten Rollen für Amazon EMR. Mit dieser Konfigurationsdatei wird der Cluster, wenn die `--use-default-roles`-Option angegeben ist, mit dem *MyCustomServiceRoleForEMR* und *MyCustomServiceRoleForClusterEC2Instances* erstellt. Standardmäßig gibt die config-Datei die standardmäßige `service_role` als `AmazonElasticMapReduceRole` und das standardmäßige `instance_profile` als `EMR_EC2_DefaultRole` an.

```
[default]  
output = json  
region = us-west-1  
aws_access_key_id = myAccessKeyID  
aws_secret_access_key = mySecretAccessKey  
emr =  
    service_role = MyCustomServiceRoleForEMR  
    instance_profile = MyCustomServiceRoleForClusterEC2Instances
```

Konfigurieren von IAM-Rollen für EMRFS-Anforderungen an Amazon S3

Note

Die auf dieser Seite beschriebene EMRFS-Rollenzuordnungsfunktion wurde mit der Einführung von Amazon S3 Access Grants in Amazon EMR 6.15.0 verbessert. Für eine skalierbare Zugriffskontrolllösung für Ihre Daten in Amazon S3 empfehlen wir, [S3 Access Grants mit Amazon EMR](#) zu verwenden.

Wenn eine Anwendung, die auf einem -Cluster ausgeführt wird, auf Daten mithilfe des `s3://mydata`-Formats verweist, wird EMRFS von Amazon EMR verwendet, um die Anforderung zu stellen. Um mit Amazon S3 zu interagieren, geht EMRFS von den Berechtigungsrichtlinien aus, die mit Ihrem [EC2 Amazon-Instance-Profil](#) verknüpft sind. Es wird dasselbe EC2 Amazon-Instance-Profil verwendet, unabhängig davon, welcher Benutzer oder welche Gruppe die Anwendung ausführt, oder vom Speicherort der Daten in Amazon S3.

Wenn Sie Cluster mit mehreren Benutzern haben, die unterschiedliche Zugriffsebenen auf Daten in Amazon S3 über EMRFS benötigen, können Sie eine Sicherheitskonfiguration mit IAM-Rollen für EMRFS einrichten. EMRFS kann je nach Benutzer oder Gruppe, die die Anfrage stellt, oder basierend auf dem Speicherort der Daten in Amazon S3 eine andere Servicerolle für EC2 Cluster-Instances übernehmen. Jede IAM-Rolle für EMRFS kann andere Berechtigungen für den Zugriff auf Daten in Amazon S3 besitzen. Weitere Informationen zur Servicerolle für EC2 Cluster-Instances finden Sie unter [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#).

Die Verwendung benutzerdefinierter IAM-Rollen für EMRFS wird in Amazon-EMR-Versionen 5.10.0 und höher unterstützt. Wenn Sie eine ältere Version verwenden oder zusätzliche Berechtigungsanforderungen haben, die über das, was IAM-Rollen für EMRFS bieten, hinausgehen, können Sie stattdessen einen benutzerdefinierten Anmeldeinformationsanbieter erstellen. Weitere Informationen finden Sie unter [Autorisieren des Zugriffs auf die EMRFS Daten in Amazon S3](#).

Wenn Sie eine Sicherheitskonfiguration nutzen, um IAM-Rollen für EMRFS anzugeben, richten Sie Rollenzuordnungen ein. Jede Rollenzuordnung gibt eine IAM-Rolle an, die Kennungen entspricht. Diese Kennungen bestimmen die Basis für den Zugriff auf Amazon S3 über EMRFS. Die Kennungen können Benutzer, Gruppen oder Amazon-S3-Präfixe sein, die einen Datenspeicherort angeben. Wenn EMRFS eine Anfrage an Amazon S3 stellt und die Anfrage mit der Grundlage für den Zugriff übereinstimmt, lässt EMRFS EC2 Cluster-Instances die entsprechende IAM-Rolle für die Anfrage

übernehmen. Die mit dieser Rolle verknüpften IAM-Berechtigungen gelten anstelle der IAM-Berechtigungen, die der Servicerolle für Cluster-Instances zugewiesen sind. EC2

Die Benutzer und Gruppen in einer Rollenzuordnung sind Hadoop-Benutzer und -gruppen, die auf dem Cluster definiert sind. Benutzer und Gruppen werden EMRFS im Kontext der Anwendung übergeben, die es verwendet (z. B. YARN-Benutzer-Identitätswechsel). Das Amazon-S3-Präfix kann ein Bucket-Spezifizierer beliebiger Tiefe sein (z. B. `s3://amzn-s3-demo-bucket` oder `s3://amzn-s3-demo-bucket/myproject/mydata`). Sie können mehrere Kennungen in einer einzigen Rollenzuordnung angeben, die jedoch alle vom selben Typ sein müssen.

 Important

IAM-Rollen für EMRFS bieten die Isolierung auf Anwendungsebene zwischen Benutzern der Anwendung. Sie bieten keine Isolierung auf Host-Ebene zwischen Benutzern auf dem Host. Jeder Benutzer mit Zugriff auf das Cluster kann die Isolation umgehen, um eine Rolle zu übernehmen.

Wenn eine Cluster-Anwendung über EMRFS eine Anforderung an Amazon S3 stellt, wertet EMRFS Rollenzuordnungen in der Reihenfolge von oben nach unten aus, in der sie in der Sicherheitskonfiguration erscheinen. Wenn eine über EMRFS gestellte Anfrage mit keiner Kennung übereinstimmt, verwendet EMRFS die Servicerolle für Clusterinstanzen. EC2 Aus diesem Grund empfehlen wir, dass Sie die Richtlinien, die dieser Rolle zugeordnet werden, auf Berechtigungen in Amazon S3 begrenzen. Weitere Informationen finden Sie unter [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#).

Konfigurieren von -Rollen

Bevor Sie eine Sicherheitskonfiguration mit IAM-Rollen für EMRFS einrichten, müssen Sie die Rollen und die Berechtigungsrichtlinien für die Rollen planen und erstellen. Weitere Informationen finden Sie unter [Wie funktionieren Rollen](#) für Instanzen? EC2 im IAM-Benutzerhandbuch. Wir empfehlen, bei der Erstellung von Berechtigungsrichtlinien mit der verwalteten Richtlinie zu beginnen, die der Amazon EMR-Standardrolle für zugeordnet ist EC2, und diese Richtlinie dann Ihren Anforderungen entsprechend zu bearbeiten. Der standardmäßige Rollename ist `EMR_EC2_DefaultRole`, und die zu bearbeitende verwaltete Standardrichtlinie ist `AmazonElasticMapReduceforEC2Role`. Weitere Informationen finden Sie unter [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#).

Aktualisieren von Vertrauensrichtlinien, um Rollenberechtigungen zu übernehmen

Jede Rolle, die EMRFS verwendet, muss über eine Vertrauensrichtlinie verfügen, die es der Amazon EMR-Rolle des Clusters ermöglicht, EC2 diese zu übernehmen. Ebenso EC2 muss für die Amazon EMR-Rolle des Clusters eine Vertrauensrichtlinie gelten, die es EMRFS-Rollen ermöglicht, sie zu übernehmen.

Das folgende Beispiel einer Vertrauensrichtlinie ist den Rollen für EMRFS zugeordnet. Die Anweisung ermöglicht es der Amazon EMR-Standardrolle für EC2, die Rolle zu übernehmen. Beispiel: Sie verfügen über die zwei fiktiven EMRFS-Rollen `EMRFSRole_First` und `EMRFSRole_Second`. In diesem Fall wird diese Richtlinienanweisung den Vertrauensrichtlinien für jede davon hinzugefügt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam:::role/EMR_EC2_DefaultRole"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Außerdem wird im folgenden Beispiel die Vertrauensrichtlinienanweisung der `EMR_EC2_DefaultRole` hinzugefügt, um zu erlauben, dass die beiden fiktiven EMRFS-Rollen diese übernehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam:::role/EMRFSRole_First",
          "arn:aws:iam:::role/EMRFSRole_Second"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
]
}
```

So aktualisieren Sie die Vertrauensrichtlinie einer IAM-Rolle

Öffnen Sie unter <https://console.aws.amazon.com/iam/> die IAM-Konsole.

1. Wählen Sie Roles (Rollen), geben Sie den Namen der Rolle unter Search (Suche) ein und wählen Sie dann Role name (Rollenname) aus.
2. Wählen Sie auf der Registerkarte Trust Relationships (Vertrauensbeziehungen) Edit Trust Relationship (Vertrauensbeziehung bearbeiten) aus.
3. Fügen Sie eine Vertrauensanweisung gemäß dem Richtliniendokument und den Richtlinien oben hinzu und wählen Sie dann Vertrauensrichtlinie updaten.

Angeben einer Rolle als Schlüsselbenutzer

Wenn eine Rolle den Zugriff auf einen Speicherort in Amazon S3 zulässt, der mit einem AWS KMS key verschlüsselt ist, muss die Rolle als Schlüsselbenutzer angegeben werden. So erhält die Rolle die Berechtigung, den KMS-Schlüssel zu verwenden. Weitere Informationen finden Sie unter [Schlüsselrichtlinien in AWS KMS](#) im Entwicklerhandbuch für AWS Key Management Service .

Einrichten einer Sicherheitskonfiguration mit IAM-Rollen für EMRFS

 Important

Wenn keine der von Ihnen angegebenen IAM-Rollen für EMRFS zutrifft, greift EMRFS auf die Amazon EMR-Rolle für zurück. EC2 Sie sollten die Berechtigungen dieser Rolle auf Amazon S3 einschränken wie für Ihre Anwendung erforderlich, und dann diese benutzerdefinierte Rolle anstelle von EMR_EC2_DefaultRole angeben, wenn Sie einen Cluster erstellen. Weitere Informationen erhalten Sie unter [Passen Sie IAM-Rollen mit Amazon EMR an](#) und [Angabe benutzerdefinierter IAM-Rollen beim Erstellen eines Clusters](#).

So geben Sie IAM-Rollen für EMRFS-Anforderungen an Amazon S3 mithilfe der Konsole an

1. Erstellen Sie eine Sicherheitskonfiguration, die Rollenzuordnungen spezifiziert:
 - a. Wählen Sie in der Amazon-EMR-Konsole die Optionen Sicherheitskonfigurationen und Erstellen aus.

- b. Geben Sie in Name (Name) einen Namen für die Sicherheitskonfiguration ein. Verwenden Sie diesen Namen zum Angeben der Sicherheitskonfiguration, wenn Sie einen Cluster erstellen.
 - c. Wählen Sie IAM-Rollen für EMRFS-Anforderungen an Amazon S3 verwenden aus.
 - d. Wählen Sie eine anzuwendende IAM-Rolle aus. Wählen Sie zudem unter Grundlage für Zugriff einen ID-Typ (Benutzer, Gruppen oder S3-Präfixen) aus der Liste aus und geben Sie die entsprechenden Kennungen ein. Wenn Sie mehrere Kennungen verwenden, trennen Sie diese durch Komma (ohne Leerzeichen dazwischen) voneinander ab. Weitere Informationen zu den einzelnen ID-Typen finden Sie unten in der [JSON configuration reference](#).
 - e. Wählen Sie Add role (Rolle hinzufügen) aus, um zusätzliche Rollenzuordnungen einzurichten wie im vorherigen Schritt beschrieben.
 - f. Richten Sie weitere Sicherheitskonfigurationsoptionen ein wie erforderlich. Wählen Sie Create (Erstellen) aus. Weitere Informationen finden Sie unter [Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI](#).
2. Geben Sie die Sicherheitskonfiguration an, die Sie oben erstellt haben, wenn Sie einen Cluster erstellen. Weitere Informationen finden Sie unter [Geben Sie eine Sicherheitskonfiguration für einen Amazon EMR-Cluster an](#).

Um IAM-Rollen für EMRFS-Anfragen an Amazon S3 anzugeben, verwenden Sie den AWS CLI

1. Verwenden Sie den Befehl `aws emr create-security-configuration`. Dadurch wird ein Name für die Sicherheitskonfiguration spezifiziert sowie die Sicherheitskonfigurationsdetails im JSON-Format.

Der unten gezeigte Beispielbefehl erstellt eine Sicherheitskonfiguration namens `EMRFS_Roles_Security_Configuration`. Diese basiert auf einer JSON-Struktur aus der Datei `MyEmrfsSecConfig.json`, die in demselben Verzeichnis gespeichert ist, in dem der Befehl ausgeführt wird.

```
aws emr create-security-configuration --name EMRFS_Roles_Security_Configuration --  
security-configuration file://MyEmrfsSecConfig.json.
```

Verwenden Sie die folgenden Richtlinien für die Struktur der Datei `MyEmrfsSecConfig.json`. Sie können diese Struktur zusammen mit Strukturen für andere Sicherheitskonfigurationsoptionen angeben. Weitere Informationen finden Sie unter [Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI](#).

Im Folgenden finden Sie ein JSON-Beispiel für die Angabe benutzerdefinierter IAM-Rollen für EMRFS innerhalb einer Sicherheitskonfiguration. Es zeigt Rollenzuordnungen für die drei verschiedenen Identifier-Typen, gefolgt von einer Parameterreferenz.

```
{
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_demo_s3_buckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://amzn-s3-demo-bucket1/", "s3://amzn-s3-demo-bucket2/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

Parameter	Beschreibung
"AuthorizationConfiguration":	Erforderlich
"EmrFsConfiguration":	Erforderlich Enthält Rollenzuordnungen.

Parameter	Beschreibung
"RoleMappings":	Erforderlich Enthält eine oder mehrere Rollenzuordnungsdefinitionen. Rollenzuordnungen werden in der Reihenfolge bewertet, in der sie von oben nach unten angezeigt werden. Wenn eine Rollenzuweisung für einen EMRFS-Datenaufruf in Amazon S3 als wahr bewertet wird, werden keine weiteren Rollenzuordnungen ausgewertet und EMRFS verwendet die angegebene IAM-Rolle für die Anfrage. Rollenzuordnungen bestehen aus den folgenden erforderlichen Parametern:
"Role":	Gibt den ARN-Bezeichner einer IAM-Rolle im Format <code>arn:aws:iam:: <i>account-id</i> :role/<i>role-name</i></code> an. Dies ist die IAM-Rolle, die Amazon EMR übernimmt, wenn die EMRFS-Anfrage an Amazon S3 mit einer der angegebenen Identifiers übereinstimmt.

Parameter	Beschreibung
"IdentifierType":	Kann einer der folgenden sein: "User" gibt an, dass es sich bei den Kennungen um einen oder mehrere Hadoop-Benutzer handelt, bei denen es sich um Linux-Kontob Benutzer oder Kerberos-Prinzipale handeln kann. Wenn die EMRFS-Anfrage von dem oder den angegebenen Benutzern stammt, wird die IAM-Rolle übernommen. "Prefix" gibt an, dass der Identifier ein Amazon-S3-Speicherort ist. Die IAM-Rolle wird für Anrufe an den Standort oder die Standorte mit den angegebenen Präfixen übernommen. Das Präfix <code>s3://amzn-s3-demo-bucket/</code> entspricht beispielsweise <code>s3://amzn-s3-demo-bucket/mydir</code> und <code>s3://amzn-s3-demo-bucket/anotherdir</code>. "Group" gibt an, dass es sich bei den Identifikatoren um eine oder mehrere Hadoop-Gruppen handelt. Die IAM-Rolle wird übernommen, wenn die Anfrage von einem Benutzer in der oder den angegebenen Gruppen stammt.
"Identifiers":	Gibt einen oder mehrere Kennungen des entsprechenden Kennungstyps an. Trennen Sie mehrere Bezeichner durch Kommas ohne Leerzeichen.

- Verwenden Sie den Befehl `aws emr create-cluster`, um einen Cluster einzurichten, und geben Sie die Sicherheitskonfiguration an, die Sie im vorherigen Schritt erstellt haben.

Im folgenden Beispiel wird ein Cluster erstellt, bei dem Standard-Core-Hadoop-Anwendungen installiert sind. Der Cluster verwendet die oben erstellte Sicherheitskonfiguration als `EMRFS_Roles_Security_Configuration` und verwendet auch eine benutzerdefinierte Amazon EMR-Rolle für EC2, `EC2_Role_EMR_Restrict_S3`, die mit dem `InstanceProfile` Argument des `--ec2-attributes` Parameters angegeben wird.

 Note

Linux-Zeilenfortsetzungszeichen (`\`) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (`^`).

```
aws emr create-cluster --name MyEmrFsS3RolesCluster \  
--release-label emr-7.9.0 --ec2-attributes  
  InstanceProfile=EC2_Role_EMR_Restrict_S3,KeyName=MyKey \  
--instance-type m5.xlarge --instance-count 3 \  
--security-configuration EMRFS_Roles_Security_Configuration
```

Ressourcenbasierte Richtlinien für den Zugriff von Amazon EMR auf AWS Glue Data Catalog verwenden

Wenn Sie AWS Glue in Verbindung mit Hive, Spark oder Presto in Amazon EMR verwenden, unterstützt AWS Glue ressourcenbasierte Richtlinien zur Steuerung des Zugriffs auf Datenkatalogressourcen. Zu diesen Ressourcen gehören Datenbanken, Tabellen, Verbindungen und benutzerdefinierte Funktionen. Weitere Informationen finden Sie unter [Verwenden von ressourcenbasierten Richtlinien für AWS Glue](#) im AWS -Glue-Entwicklerhandbuch.

Wenn Sie ressourcenbasierte Richtlinien verwenden, um den Zugriff auf AWS Glue von Amazon EMR aus zu beschränken, muss der Principal, den Sie in der Berechtigungsrichtlinie angeben, der Rollen-ARN sein, der dem EC2 Instance-Profil zugeordnet ist, das bei der Erstellung eines Clusters angegeben wird. Für eine ressourcenbasierte Richtlinie, die an einen Katalog angehängt ist, können Sie beispielsweise den Rollen-ARN für die Standarddienstrolle für EC2 Clusterinstanzen angeben, `EMR_EC2_DefaultRole` indem Sie das Principal im folgenden Beispiel gezeigte Format verwenden:

```
arn:aws:iam::acct-id:role/EMR_EC2_DefaultRole
```

Die *acct-id* kann sich von der AWS Glue-Konto-ID unterscheiden. Dies ermöglicht den Zugriff von EMR-Clustern in verschiedenen Konten aus. Sie können mehrere Principals angeben, von denen jeder aus einem anderen Konto stammt.

IAM-Rollen mit Anwendungen verwenden, die AWS -Services direkt aufrufen

Anwendungen, die auf den EC2 Instanzen eines Clusters ausgeführt werden, können das EC2 Instanzprofil verwenden, um beim Aufrufen von AWS Diensten temporäre Sicherheitsanmeldeinformationen abzurufen.

Die Hadoop-Versionen verfügbar mit Amazon EMR Version 2.3.0 und höher wurden bereits aktualisiert, um IAM-Rollen zu nutzen. Wenn Ihre Anwendung ausschließlich auf der Hadoop-Architektur ausgeführt wird und keinen Dienst direkt aufruft AWS, sollte sie ohne Änderung mit IAM-Rollen funktionieren.

Wenn Ihre Anwendung Dienste AWS direkt aufruft, müssen Sie sie aktualisieren, um die Vorteile der IAM-Rollen nutzen zu können. Das bedeutet, dass Ihre Anwendung, anstatt Kontoanmeldeinformationen von `/etc/hadoop/conf/core-site.xml` den EC2 Instances im Cluster abzurufen, ein SDK verwendet, um mithilfe von IAM-Rollen auf die Ressourcen zuzugreifen, oder die EC2 Instance-Metadaten aufruft, um die temporären Anmeldeinformationen abzurufen.

Um mithilfe eines AWS SDK auf Ressourcen mit IAM-Rollen zuzugreifen

- In den folgenden Themen wird gezeigt, wie Sie mehrere der Rollen verwenden AWS SDKs , um mithilfe von IAM-Rollen auf temporäre Anmeldeinformationen zuzugreifen. Jedes Thema beginnt mit einer Anwendungsversion, die keine IAM-Rollen verwendet, und führt Sie anschließend schrittweise durch die Konvertierung der Anwendung, um sie auf die Verwendung der IAM-Rollen vorzubereiten.
 - [Verwenden von IAM-Rollen für EC2 Amazon-Instances mit dem SDK for Java](#) im AWS SDK für Java Developer Guide
 - [Verwenden von IAM-Rollen für EC2 Amazon-Instances mit dem SDK for .NET](#) im AWS SDK für .NET Developer Guide
 - [Verwenden von IAM-Rollen für EC2 Amazon-Instances mit dem SDK for PHP](#) im AWS SDK für PHP Developer Guide

- [Verwenden von IAM-Rollen für EC2 Amazon-Instances mit dem SDK for Ruby](#) im AWS SDK für Ruby Developer Guide

Um temporäre Anmeldeinformationen aus EC2 Instance-Metadaten abzurufen

- Rufen Sie die folgende URL von einer EC2 Instance aus auf, die mit der angegebenen IAM-Rolle ausgeführt wird. Diese gibt die zugehörigen temporären Sicherheitsanmeldedaten (AccessKeyId, SecretAccessKey SessionToken, und Expiration) zurück. Das folgende Beispiel verwendet das Standard-Instance-Profil für Amazon EMR, `EMR_EC2_DefaultRole`.

```
GET http://169.254.169.254/latest/meta-data/iam/security-credentials/EMR_EC2_DefaultRole
```

Weitere Informationen zum Schreiben von Anwendungen, die IAM-Rollen verwenden, finden Sie unter [Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden, Zugriff auf AWS Ressourcen gewähren](#).

Weitere Informationen zu temporären Sicherheitsanmeldeinformationen finden Sie unter [Verwenden temporärer Sicherheitsanmeldeinformationen](#) im Handbuch Verwenden temporärer Sicherheitsanmeldeinformationen.

Benutzern und Gruppen gestatten, Rollen zu erstellen und zu ändern

IAM-Prinzipale (Benutzer und Gruppen), die Rollen für einen Cluster erstellen, ändern und festlegen, einschließlich Standardrollen, müssen die Berechtigung haben, die folgenden Aktionen durchzuführen. Weitere Informationen zu den einzelnen Aktionen finden Sie unter [Aktionen](#) in der IAM-API-Referenz.

- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:AddRoleToInstanceProfile`
- `iam:ListRoles`
- `iam:GetPolicy`

- `iam:GetInstanceProfile`
- `iam:GetPolicyVersion`
- `iam:AttachRolePolicy`
- `iam:PassRole`

Die Berechtigung `iam:PassRole` gewährt die Erstellung von Clustern. Die restlichen Berechtigungen gewähren die Erstellung von Standardrollen.

Weitere Informationen zum Zuweisen von Berechtigungen in IAM finden Sie unter [Ändern von Berechtigungen für einen IAM-Benutzer](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Amazon-EMR-Richtlinien

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon-EMR-Ressourcen. Sie können auch keine Aufgaben mit der AWS Management Console, AWS CLI, oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von Richtlinien auf der JSON-Registerkarte](#) im IAM-Benutzerhandbuch.

Themen

- [Bewährte Methoden für Richtlinien für Amazon EMR](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Verwaltete Richtlinien von Amazon EMR](#)
- [IAM-Richtlinien für Tag-basierten Zugriff auf Cluster und EMR-Notebooks](#)
- [Die `ModifyInstanceGroup` Aktion in Amazon EMR ablehnen](#)
- [Fehlerbehebung für Amazon-EMR-Identität und -Zugriff](#)

Bewährte Methoden für Richtlinien für Amazon EMR

Identitätsbasierte Richtlinien sind sehr leistungsfähig. Damit wird bestimmt, ob jemand Amazon-EMR-Ressourcen in Ihrem Konto erstellen, löschen oder darauf zugreifen kann. Für diese Aktionen können

Kosten für Ihr AWS Konto anfallen. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- **Erste Schritte mit AWS verwalteten Richtlinien** — Um schnell mit der Nutzung von Amazon EMR zu beginnen, verwenden Sie AWS verwaltete Richtlinien, um Ihren Mitarbeitern die erforderlichen Berechtigungen zu erteilen. Diese Richtlinien sind bereits in Ihrem Konto verfügbar und werden von AWS. Weitere Informationen finden [Sie unter Erste Schritte zur Nutzung von Berechtigungen mit AWS verwalteten Richtlinien](#) im IAM-Benutzerhandbuch und. [Verwaltete Richtlinien von Amazon EMR](#)
- **Gewähren Sie die geringstmöglichen Berechtigungen** – Gewähren Sie beim Erstellen benutzerdefinierter Richtlinien nur die Berechtigungen, die zum Ausführen einer Aufgabe erforderlich sind. Beginnen Sie mit einem Minimalsatz von Berechtigungen und gewähren Sie zusätzliche Berechtigungen wie erforderlich. Dies ist sicherer, als mit Berechtigungen zu beginnen, die zu weit gefasst sind, und dann später zu versuchen, sie zu begrenzen. Weitere Informationen finden Sie unter [Gewähren von geringsten Rechten](#) im IAM-Benutzerhandbuch.
- **Aktivieren Sie für sensible Vorgänge MFA** – Fordern Sie von Benutzern die Verwendung von Multi-Faktor-Authentifizierung (MFA), um zusätzliche Sicherheit beim Zugriff auf sensible Ressourcen oder API-Operationen zu bieten. Weitere Informationen finden Sie unter [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.
- **Verwenden Sie Richtlinienbedingungen, um zusätzliche Sicherheit zu bieten** – Definieren Sie die Bedingungen, unter denen Ihre identitätsbasierten Richtlinien den Zugriff auf eine Ressource zulassen, soweit praktikabel. Beispielsweise können Sie Bedingungen schreiben, die eine Reihe von zulässigen IP-Adressen festlegen, von denen eine Anforderung stammen muss. Sie können auch Bedingungen schreiben, die Anforderungen nur innerhalb eines bestimmten Datums- oder Zeitbereichs zulassen oder die Verwendung von SSL oder MFA fordern. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die 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 OR-API. AWS

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

```

```

{
  "Sid": "ViewOwnUserInfo",
  "Effect": "Allow",
  "Action": [
    "iam:GetUser",
    "iam:GetUserPolicy",
    "iam:ListAttachedUserPolicies",
    "iam:ListGroupsForUser",
    "iam:ListUserPolicies"
  ],
  "Resource": [
    "arn:aws:iam::*:user/${aws:username}"
  ]
},
{
  "Sid": "NavigateInConsole",
  "Effect": "Allow",
  "Action": [
    "iam:GetGroupPolicy",
    "iam:GetPolicy",
    "iam:GetPolicyVersion",
    "iam:ListAttachedGroupPolicies",
    "iam:ListGroupPolicies",
    "iam:ListGroups",
    "iam:ListPolicies",
    "iam:ListPolicyVersions",
    "iam:ListUsers"
  ],
  "Resource": "*"
}
]
}

```

Verwaltete Richtlinien von Amazon EMR

Die einfachste Möglichkeit, Vollzugriff oder Lesezugriff auf benötigte Amazon-EMR-Aktionen zu erteilen, besteht in der Verwendung von IAM-verwalteten Richtlinien für Amazon EMR. Verwaltete Richtlinien haben den Vorteil, automatisch aktualisiert zu werden, wenn sich die Berechtigungsanforderungen ändern. Wenn Sie eingebundene Richtlinien verwenden, können Service-Veränderungen auftreten, die zu Berechtigungsfehlern führen.

Amazon EMR wird bestehende verwaltete Richtlinien (v1-Richtlinien) zugunsten neuer verwalteter Richtlinien (v2-Richtlinien) ablehnen. Die neuen verwalteten Richtlinien wurden detailliert auf

bewährte Verfahren abgestimmt. AWS Sobald die bestehenden verwalteten Richtlinien der Version 1 veraltet sind, können Sie diese Richtlinien keinen neuen IAM-Rollen oder -Benutzern mehr zuordnen. Bestehende Rollen und Benutzer, die veraltete Richtlinien verwenden, können diese weiterhin verwenden. Die verwalteten v2-Richtlinien schränken den Zugriff mithilfe von Tags ein. Sie lassen nur bestimmte Amazon-EMR-Aktionen zu und erfordern Cluster-Ressourcen, die mit einem EMR-spezifischen Schlüssel gekennzeichnet sind. Wir empfehlen Ihnen, die Dokumentation sorgfältig zu lesen, bevor Sie die neuen v2-Richtlinien verwenden.

Die v1-Richtlinien werden in der Liste der Richtlinien der IAM-Konsole mit einem Warnsymbol daneben als veraltet markiert. Die veralteten Richtlinien werden die folgenden Merkmale aufweisen:

- Sie werden unverändert für alle gegenwärtig angefügten Benutzer, Gruppen und Rollen funktionsfähig. Alles funktioniert normal.
- Sie können nicht neuen Benutzern, Gruppen oder Rollen angefügt werden. Wenn Sie eine der Richtlinien von einer gegenwärtigen Entität trennen, können Sie sie nicht wieder anfügen.
- Nachdem Sie eine v1-Richtlinie von allen aktuellen Entitäten getrennt haben, ist die Richtlinie nicht mehr sichtbar und kann nicht mehr verwendet werden.

In der folgenden Tabelle werden die Änderungen zwischen den aktuellen Richtlinien (v1) und v2-Richtlinien zusammengefasst.

Von Amazon EMR verwaltete Richtlinienänderungen

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
Standard-EMR-Servicerolle und angehängte verwaltete Richtlinie	Rollename: EMR_DefaultRole V1-Richtlinie (wird nicht mehr unterstützt): AmazonElasticMapReduceRole(EMR-Servicerolle) V2-Richtliniennamen (mit eingeschr	Ermöglicht Amazon EMR, in Ihrem Namen andere AWS Services aufzurufen, wenn Ressourcen bereitgestellt und Service-Level-Aktionen ausgeführt werden. Diese Rolle ist für alle Cluster erforderlich.	Die Richtlinie fügt die neue Berechtigung hinzu. "ec2:DescribeInstances" Dieser API-Vorgang gibt eine Liste von Instanztypen zurück, die von einer Liste der angegebenen

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
	änktem Geltungsbereich): AmazonEMR ServicePolicy_v2		Availability Zones unterstützt werden.

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
Von IAM verwaltete Richtlinie für vollen Zugriff durch angehängte Benutzer, Rollen oder Gruppen	V2-Richtliniennamen (mit Geltungsbereich): AmazonEMRServicePolicy_v2	Erlaubt Benutzern volle Berechtigungen für EMR-Aktionen. Beinhaltet iam: PassRole Berechtigungen für Ressourcen.	Die Richtlinie setzt voraus, dass Benutzer Benutzer-Tags zu Ressourcen hinzufügen müssen, bevor sie diese Richtlinie verwenden können. Siehe Taggen von Ressourcen zur Verwendung verwalteter Richtlinien. Für die PassRole Aktion iam: muss die PassedToService Bedingung iam: auf den angegebenen Dienst gesetzt sein. Der Zugriff auf Amazon EC2, Amazon S3 und andere Dienste ist standardmäßig nicht erlaubt. Weitere Informationen finden Sie unter Verwaltete IAM-Richtlinie für vollen Zugriff (verwaltete Standardrichtlinie v2).

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
Von IAM verwaltete Richtlinie für vollständigen EMR-Zugriff durch angehängte Benutzer, Rollen oder Gruppen	V1-Richtlinie (wird veraltet): AmazonElasticMapReduceReadOnlyAccess V2-Richtliniennamen (mit Geltungsbereich): AmazonEMRReadOnlyAccessPolicy_v2	Ermöglicht Benutzern nur Leseberechtigungen für Amazon-EMR-Aktionen.	Mit Berechtigungen können nur bestimmte schreibgeschützte ElasticMapReduce-Aktionen ausgeführt werden. Der Zugriff auf Amazon S3 ist standardmäßig nicht zulässig. Weitere Informationen finden Sie unter Verwaltete IAM-Richtlinie für vollen Zugriff (verwaltete Standardrichtlinie v2)

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
Standard-EMR-Servicerolle und angehängte verwaltete Richtlinie	Rollename: EMR_DefaultRole V1-Richtlinie (wird nicht mehr unterstützt): AmazonElasticMapReduceRole(EMR-Servicerolle) V2-Richtliniename (mit eingeschränktem Geltungsbereich): AmazonEMRServicePolicy_v2	Ermöglicht Amazon EMR, in Ihrem Namen andere AWS Services aufzurufen, wenn Ressourcen bereitgestellt und Service-Level-Aktionen ausgeführt werden. Diese Rolle ist für alle Cluster erforderlich.	Die v2-Servicerolle und die v2-Standardrichtlinie ersetzen die veraltete Rolle und Richtlinie. Die Richtlinie setzt voraus, dass Benutzer Benutzer-Tags zu Ressourcen hinzufügen müssen, bevor sie diese Richtlinie verwenden können. Siehe Taggen von Ressourcen zur Verwendung verwalteter Richtlinien. Siehe Servicerolle für Amazon EMR (EMR-Rolle).

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
Service-Rolle für EC2 Cluster-Instances (Instance-Profil) EC2	Rollename: EMR_ _ EC2_DefaultRole Veralteter Richtliniename: Rolle AmazonElasticMapReducefor EC2	Ermöglicht Anwendungen, die auf einem EMR-Cluster ausgeführt werden, auf andere AWS-Ressourcen wie Amazon S3 zuzugreifen. Wenn Sie beispielsweise Apache-Spark-Aufträge ausführen, die Daten von Amazon S3 verarbeiten, muss die Richtlinie den Zugriff auf solche Ressourcen zulassen.	Sowohl die Standardrolle als auch die Standardrichtlinie werden demnächst veraltet sein. Es gibt keinen Ersatz für die verwaltete AWS Standardrolle oder -richtlinie. Sie müssen eine ressourcen- oder identitätsbasierte Richtlinie bereitstellen. Das bedeutet, dass Anwendungen, die auf einem EMR-Cluster ausgeführt werden, standardmäßig keinen Zugriff auf Amazon S3 oder andere Ressourcen haben, es sei denn, Sie fügen diese manuell zur Richtlinie hinzu. Siehe Standardrolle und verwaltete Richtlinie.

Richtlinientyp	Richtliniennamen	Zweck der Richtlinie	Änderungen der v2-Richtlinie
Andere Richtlinien EC2 für Servicerollen	Aktuelle Richtliniennamen: AmazonElasticMapReduceforAutoScalingRole, AmazonElasticMapReduceEditorsRole, EMRCleanup Amazon-Richtlinie	Stellt Berechtigungen bereit, die Amazon EMR benötigt, um auf andere AWS Ressourcen zuzugreifen und Aktionen auszuführen, wenn Auto Scaling, Notebooks oder EC2 Ressourcen bereinigt werden.	Keine Änderungen für Version 2.

Sicherung von iam: PassRole

Die verwalteten Standardrichtlinien von Amazon EMR mit vollen Berechtigungen beinhalten `iam:PassRole`-Sicherheitskonfigurationen, darunter die folgenden:

- `iam:PassRole`-Berechtigungen nur für bestimmte Amazon-EMR-Standardrollen.
- `iam:PassedToService` Bedingungen, die es Ihnen ermöglichen, die Richtlinie nur mit bestimmten AWS Diensten zu verwenden, z. B. `elasticmapreduce.amazonaws.com` und `ec2.amazonaws.com`.

Sie können die JSON-Version der [Amazon EMRFull AccessPolicy _v2- und Amazon EMRService Policy_v2-Richtlinien in der IAM-Konsole](#) anzeigen. Wir empfehlen, dass Sie neue Cluster mit den verwalteten v2-Richtlinien erstellen.

Wenn Sie benutzerdefinierte Richtlinien erstellen müssen, empfehlen wir Ihnen, mit verwalteten Richtlinien zu beginnen und diese entsprechend Ihren Anforderungen zu bearbeiten.

Informationen zum Anfügen von Richtlinien an [Benutzer \(Prinzipale\) finden Sie unter Arbeiten mit verwalteten Richtlinien über die AWS Management Console](#) im IAM-Benutzerhandbuch.

Taggen von Ressourcen zur Verwendung verwalteter Richtlinien

Amazon EMRService Policy_v2 und Amazon EMRFull AccessPolicy _v2 hängen vom begrenzten Zugriff auf Ressourcen ab, die Amazon EMR bereitstellt oder verwendet. Der eingeschränkte

Umfang wird dadurch erreicht, dass der Zugriff nur auf die Ressourcen beschränkt wird, denen ein vordefiniertes Benutzer-Tag zugeordnet ist. Wenn Sie eine dieser beiden Richtlinien verwenden, müssen Sie bei der Bereitstellung des Clusters das vordefinierte Benutzer-Tag `for-use-with-amazon-emr-managed-policies = true` übergeben. Amazon EMR verbreitet diese Tags automatisch. Darüber hinaus müssen Sie den im folgenden Abschnitt aufgelisteten Ressourcen ein Benutzer-Tag hinzufügen. Wenn Sie die Amazon-EMR-Konsole verwenden, um Ihren Cluster zu starten, finden Sie weitere Informationen unter [Überlegungen zur Verwendung der Amazon-EMR-Konsole zum Starten von Clustern mit verwalteten v2-Richtlinien](#).

Um verwaltete Richtlinien zu verwenden, übergeben Sie das Benutzer-Tag `for-use-with-amazon-emr-managed-policies = true`, wenn Sie einen Cluster mit der CLI, dem SDK oder einer anderen Methode bereitstellen.

Wenn Sie das Tag übergeben, leitet Amazon EMR das Tag an die privaten Subnetz-ENI-, EC2 Instance- und EBS-Volumes weiter, die es erstellt. Amazon EMR kennzeichnet auch automatisch Sicherheitsgruppen, die es erstellt. Wenn Sie jedoch möchten, dass Amazon EMR mit einer bestimmten Sicherheitsgruppe gestartet wird, müssen Sie sie taggen. Für Ressourcen, die nicht von Amazon EMR erstellt wurden, müssen Sie diesen Ressourcen Tags hinzufügen. Beispielsweise müssen Sie EC2 Amazon-Subnetze, EC2 Sicherheitsgruppen (sofern sie nicht von Amazon EMR erstellt wurden) und VPCs (wenn Amazon EMR Sicherheitsgruppen erstellen soll) taggen. Um Cluster mit verwalteten v2-Richtlinien zu starten VPCs, müssen Sie diese VPCs mit dem vordefinierten Benutzertag kennzeichnen. Siehe [Überlegungen zur Verwendung der Amazon-EMR-Konsole zum Starten von Clustern mit verwalteten v2-Richtlinien](#).

Weiterverbreitetes benutzerdefiniertes Tagging

Amazon EMR kennzeichnet Ressourcen, die es mit den Amazon-EMR-Tags erstellt, die Sie bei der Erstellung eines Clusters angeben. Amazon EMR wendet Tags auf die Ressourcen an, die es während der Lebensdauer des Clusters erstellt.

Amazon EMR verbreitet Benutzer-Tags für die folgenden Ressourcen:

- Privates Subnetz-ENI (elastische Netzwerkschnittstellen für Servicezugriff)
- EC2 Instanzen
- EBS-Datenträger
- EC2 Vorlage starten

Automatisch getaggte Sicherheitsgruppen

Amazon EMR kennzeichnet EC2 Sicherheitsgruppen, die es erstellt, mit dem Tag, das für verwaltete v2-Richtlinien für Amazon EMR erforderlich ist `for-use-with-amazon-emr-managed-policies`, unabhängig davon, welche Tags Sie im Befehl `create cluster` angeben. Für eine Sicherheitsgruppe, die vor der Einführung der verwalteten Richtlinien der Version 2 erstellt wurde, kennzeichnet Amazon EMR die Sicherheitsgruppe nicht automatisch. Wenn Sie verwaltete v2-Richtlinien mit den Standardsicherheitsgruppen verwenden möchten, die bereits im Konto vorhanden sind, müssen Sie die Sicherheitsgruppen manuell mit `for-use-with-amazon-emr-managed-policies = true` taggen.

Manuell getaggte Clusterressourcen

Sie müssen einige Cluster-Ressourcen manuell taggen, damit sie über Amazon-EMR-Standardrollen abgerufen werden können.

- Sie müssen EC2 Sicherheitsgruppen und EC2 Subnetze manuell mit dem von Amazon EMR verwalteten Policy-Tag kennzeichnen. `for-use-with-amazon-emr-managed-policies`
- Sie müssen eine VPC manuell taggen, wenn Amazon EMR Standardsicherheitsgruppen erstellen soll. EMR versucht, eine Sicherheitsgruppe mit dem spezifischen Tag zu erstellen, falls die Standardsicherheitsgruppe noch nicht existiert.

Amazon EMR kennzeichnet automatisch die folgenden Ressourcen:

- Von EMR erstellte Sicherheitsgruppen EC2

Sie müssen die folgenden Ressourcen manuell taggen:

- EC2 Subnetz
- EC2 Sicherheitsgruppen

Optional können Sie die folgenden Ressourcen manuell taggen:

- VPC – nur wenn Sie möchten, dass Amazon EMR Sicherheitsgruppen erstellt

Überlegungen zur Verwendung der Amazon-EMR-Konsole zum Starten von Clustern mit verwalteten v2-Richtlinien

Sie können Cluster mit verwalteten v2-Richtlinien mithilfe der Amazon-EMR-Konsole bereitstellen. Im Folgenden finden Sie einige Überlegungen, wenn Sie die Konsole zum Starten von Amazon-EMR-Clustern verwenden.

- Sie müssen das vordefinierte Tag nicht übergeben. Amazon EMR fügt das Tag automatisch hinzu und leitet es an die entsprechenden Komponenten weiter.
- Bei Komponenten, die manuell markiert werden müssen, versucht die alte Amazon-EMR-Konsole, sie automatisch zu kennzeichnen, sofern Sie über die erforderlichen Berechtigungen zum Taggen von Ressourcen verfügen. Wenn Sie nicht berechtigt sind, Ressourcen zu taggen, oder wenn Sie die Konsole verwenden möchten, bitten Sie Ihren Administrator, diese Ressourcen zu kennzeichnen.
- Sie können Cluster mit verwalteten v2-Richtlinien nur starten, wenn alle Voraussetzungen erfüllt sind.
- Die alte Amazon-EMR-Konsole zeigt Ihnen, welche Ressourcen (VPC/Subnetze) markiert werden müssen.

Von IAM verwaltete Richtlinie für vollen Zugriff (verwaltete Standardrichtlinie v2) für Amazon EMR

Die standardmäßigen verwalteten EMR-Richtlinien mit Geltungsbereich v2 gewähren Benutzern bestimmte Zugriffsrechte. Sie benötigen ein vordefiniertes Amazon-EMR-Ressourcen-Tag und `iam:PassRole` Zustandsschlüssel für Ressourcen, die von Amazon EMR verwendet werden, wie z. B. die Subnet und die SecurityGroup, die Sie zum Starten Ihres Clusters verwenden.

Um alle erforderlichen Aktionen für Amazon EMR zuzulassen, fügen Sie die `AmazonEMRFullAccessPolicy_v2`-verwaltete Richtlinie an. Diese aktualisierte verwaltete Standardrichtlinie ersetzt die [AmazonElasticMapReduceFullAccess](#) verwaltete Richtlinie.

`AmazonEMRFullAccessPolicy_v2` hängt vom begrenzten Zugriff auf Ressourcen ab, die Amazon EMR bereitstellt oder nutzt. Wenn Sie diese Richtlinie verwenden, müssen Sie bei der Bereitstellung des Clusters das Benutzer-Tag `for-use-with-amazon-emr-managed-policies = true` übergeben. Amazon EMR verbreitet diese Tags automatisch. Darüber hinaus müssen Sie möglicherweise manuell ein Benutzer-Tag zu bestimmten Ressourcentypen hinzufügen, z. B. EC2 Sicherheitsgruppen, die nicht von Amazon EMR erstellt wurden. Weitere Informationen finden Sie unter [Taggen von Ressourcen zur Verwendung verwalteter Richtlinien](#).

Die [AmazonEMRFullAccessPolicy_v2](#)-Richtlinie schützt Ressourcen, indem sie wie folgt vorgeht:

- Erfordert, dass Ressourcen für die Clustererstellung und den Zugriff auf Amazon EMR mit dem vordefinierten Tag `for-use-with-amazon-emr-managed-policies` für verwaltete Amazon-EMR-Richtlinien gekennzeichnet werden.
- Beschränkt die `iam:PassRole`-Aktion auf bestimmte Standardrollen und den `iam:PassedToService`-Zugriff auf bestimmte Services.
- Bietet standardmäßig keinen Zugriff mehr auf Amazon EC2, Amazon S3 und andere Dienste.

Im Folgenden finden Sie den Inhalt dieser Richtlinie.

 Note

Sie können die Richtlinie auch über den Konsolenlink [AmazonEMRFullAccessPolicy_v2](#) anzeigen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RunJobFlowExplicitlyWithEMRManagedTag",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:RunJobFlow"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "ElasticMapReduceActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:AddInstanceFleet",
        "elasticmapreduce:AddInstanceGroups",
        "elasticmapreduce:AddJobFlowSteps",
```

```

    "elasticmapreduce:AddTags",
    "elasticmapreduce:CancelSteps",
    "elasticmapreduce:CreateEditor",
    "elasticmapreduce:CreatePersistentAppUI",
    "elasticmapreduce:CreateSecurityConfiguration",
    "elasticmapreduce>DeleteEditor",
    "elasticmapreduce>DeleteSecurityConfiguration",
    "elasticmapreduce:DescribeCluster",
    "elasticmapreduce:DescribeEditor",
    "elasticmapreduce:DescribeJobFlows",
    "elasticmapreduce:DescribePersistentAppUI",
    "elasticmapreduce:DescribeSecurityConfiguration",
    "elasticmapreduce:DescribeStep",
    "elasticmapreduce:DescribeReleaseLabel",
    "elasticmapreduce:GetBlockPublicAccessConfiguration",
    "elasticmapreduce:GetManagedScalingPolicy",
    "elasticmapreduce:GetPersistentAppUIPresignedURL",
    "elasticmapreduce:GetAutoTerminationPolicy",
    "elasticmapreduce:ListBootstrapActions",
    "elasticmapreduce:ListClusters",
    "elasticmapreduce:ListEditors",
    "elasticmapreduce:ListInstanceFleets",
    "elasticmapreduce:ListInstanceGroups",
    "elasticmapreduce:ListInstances",
    "elasticmapreduce:ListSecurityConfigurations",
    "elasticmapreduce:ListSteps",
    "elasticmapreduce:ListSupportedInstanceTypes",
    "elasticmapreduce:ModifyCluster",
    "elasticmapreduce:ModifyInstanceFleet",
    "elasticmapreduce:ModifyInstanceGroups",
    "elasticmapreduce:OpenEditorInConsole",
    "elasticmapreduce:PutAutoScalingPolicy",
    "elasticmapreduce:PutBlockPublicAccessConfiguration",
    "elasticmapreduce:PutManagedScalingPolicy",
    "elasticmapreduce:RemoveAutoScalingPolicy",
    "elasticmapreduce:RemoveManagedScalingPolicy",
    "elasticmapreduce:RemoveTags",
    "elasticmapreduce:SetTerminationProtection",
    "elasticmapreduce:StartEditor",
    "elasticmapreduce:StopEditor",
    "elasticmapreduce:TerminateJobFlows",
    "elasticmapreduce:ViewEventsFromAllClustersInConsole"
  ],
  "Resource": "*"

```

```

    },
    {
      "Sid": "ViewMetricsInEMRConsole",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    },
    {
      "Sid": "PassRoleForElasticMapReduce",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "arn:aws:iam::*:role/EMR_DefaultRole",
        "arn:aws:iam::*:role/EMR_DefaultRole_V2"
      ],
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
        }
      }
    },
    {
      "Sid": "PassRoleForEC2",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "ec2.amazonaws.com*"
        }
      }
    },
    {
      "Sid": "PassRoleForAutoScaling",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "application-autoscaling.amazonaws.com*"
        }
      }
    }
  ]
}

```

```

    },
    {
      "Sid": "ElasticMapReduceServiceLinkedRole",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/
elasticmapreduce.amazonaws.com*/AWSServiceRoleForEMRCleanup*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "elasticmapreduce.amazonaws.com",
            "elasticmapreduce.amazonaws.com.cn"
          ]
        }
      }
    },
  ],
  {
    "Sid": "ConsoleUIActions",
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeAccountAttributes",
      "ec2:DescribeAvailabilityZones",
      "ec2:DescribeImages",
      "ec2:DescribeKeyPairs",
      "ec2:DescribeNatGateways",
      "ec2:DescribeRouteTables",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcs",
      "ec2:DescribeVpcEndpoints",
      "s3:ListAllMyBuckets",
      "iam:ListRoles"
    ],
    "Resource": "*"
  }
]
}

```

Von IAM verwaltete Richtlinie für vollen Zugriff (demnächst veraltet)

Die verwalteten Richtlinien `AmazonElasticMapReduceFullAccess` und `AmazonEMRFullAccessPolicy_v2` AWS Identity and Access Management (IAM) gewähren alle erforderlichen Aktionen für Amazon EMR und andere Services.

⚠ Important

Die verwaltete Richtlinie `AmazonElasticMapReduceFullAccess` ist veraltet und wird nicht mehr für die Verwendung mit Amazon EMR empfohlen. Nutzen Sie stattdessen [AmazonEMRFullAccessPolicy_v2](#). Wenn der IAM-Service die v1-Richtlinie irgendwann als veraltet markiert, können Sie sie keiner Rolle zuordnen. Sie können einem Cluster jedoch eine bestehende Rolle zuordnen, auch wenn diese Rolle die veraltete Richtlinie verwendet.

Die verwalteten Standardrichtlinien von Amazon EMR mit vollen Berechtigungen beinhalten `iam:PassRole`-Sicherheitskonfigurationen, darunter die folgenden:

- `iam:PassRole`-Berechtigungen nur für bestimmte Amazon-EMR-Standardrollen.
- `iam:PassedToService` Bedingungen, die es Ihnen ermöglichen, die Richtlinie nur mit bestimmten AWS Diensten wie `elasticmapreduce.amazonaws.com` und zu verwenden. `ec2.amazonaws.com`

Sie können die JSON-Version der [Amazon EMRFull AccessPolicy_v2- und Amazon EMRService Policy_v2-Richtlinien in der IAM-Konsole](#) anzeigen. Wir empfehlen, dass Sie neue Cluster mit den verwalteten v2-Richtlinien erstellen.

Sie können den Inhalt der veralteten v1-Richtlinie unter einsehen. AWS Management Console [AmazonElasticMapReduceFullAccess](#) Die `ec2:TerminateInstances` Aktion in der Richtlinie gewährt einem Benutzer oder einer Rolle die Erlaubnis, alle EC2 Amazon-Instances zu kündigen, die mit dem IAM-Konto verknüpft sind. Dies schließt Instances ein, die nicht Teil eines EMR-Clusters sind.

Von IAM verwaltete Richtlinie für schreibgeschützten Zugriff (verwaltete Standardrichtlinie v2) für Amazon EMR

Um Amazon EMR nur Leserechte zu gewähren, fügen Sie die verwaltete Amazon `EMRReadOnlyAccessPolicy_v2`-Richtlinie bei. Diese verwaltete Standardrichtlinie ersetzt die [AmazonElasticMapReduceReadOnlyAccess](#) verwaltete Richtlinie. Der Inhalt dieser Richtlinienerklärung wird im folgenden Ausschnitt dargestellt. Im Vergleich zur `AmazonElasticMapReduceReadOnlyAccess`-Richtlinie verwendet die `AmazonEMRReadOnlyAccessPolicy_v2`-Richtlinie keine Platzhalterzeichen für das `elasticmapreduce`-Element. Stattdessen beschränkt sich die standardmäßige v2-Richtlinie auf die zulässigen `elasticmapreduce`-Aktionen.

Note

Sie können den AWS Management Console Link auch verwenden, um die Richtlinie [AmazonEMRReadOnlyAccessPolicy_v2](#) einzusehen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ElasticMapReduceActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:DescribeEditor",
        "elasticmapreduce:DescribeJobFlows",
        "elasticmapreduce:DescribeSecurityConfiguration",
        "elasticmapreduce:DescribeStep",
        "elasticmapreduce:DescribeReleaseLabel",
        "elasticmapreduce:GetBlockPublicAccessConfiguration",
        "elasticmapreduce:GetManagedScalingPolicy",
        "elasticmapreduce:GetAutoTerminationPolicy",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:ListClusters",
        "elasticmapreduce:ListEditors",
        "elasticmapreduce:ListInstanceFleets",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:ListSecurityConfigurations",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:ListSupportedInstanceTypes",
        "elasticmapreduce:ViewEventsFromAllClustersInConsole"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ViewMetricsInEMRConsole",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```

```
    }
  ]
}
```

Von IAM verwaltete Richtlinie für vollen Zugriff (demnächst veraltet)

Die AmazonElasticMapReduceReadOnlyAccess-verwaltete Richtlinie ist demnächst veraltet. Sie können diese Richtlinie nicht anhängen, wenn Sie neue Cluster starten.

AmazonElasticMapReduceReadOnlyAccess wurde durch die verwaltete Standardrichtlinie [AmazonEMRReadOnlyAccessPolicy_v2](#) von Amazon EMR ersetzt. Der Inhalt dieser Richtlinienerklärung wird im folgenden Ausschnitt dargestellt. Platzhalterzeichen für das elasticmapreduce-Element geben an, dass nur Aktionen, die mit der angegebenen Zeichenfolgen beginnen, zulässig sind. Hinweis: Da diese Richtlinie Aktionen nicht ausdrücklich verweigert, kann dennoch eine andere Richtlinienanweisung verwendet werden, um den Zugriff auf bestimmte Aktionen zu gewähren.

Note

Sie können die Richtlinie auch verwenden AWS Management Console , um die Richtlinie einzusehen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:Describe*",
        "elasticmapreduce:List*",
        "elasticmapreduce:ViewEventsFromAllClustersInConsole",
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sdb:Select",
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```



```
}
```

AWS verwaltete Richtlinie: EMRDescribe ClusterPolicyFor EMRWAL

Sie können `EMRDescribeClusterPolicyForEMRWAL` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es Amazon EMR ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen zu dieser dienstbezogenen Rolle finden Sie unter [Verwenden von serviceverknüpften Rollen mit Amazon EMR für die Write-Ahead-Protokollierung](#)

Diese Richtlinie gewährt Nur-Lese-Berechtigungen, die es dem WAL-Service für Amazon EMR ermöglichen, den Status eines Clusters zu finden und zurückzugeben. Weitere Informationen zu Amazon EMR WAL finden Sie unter [Write-Ahead-Logs \(WAL\) für Amazon EMR](#).

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `emr`— Ermöglicht Principals, den Cluster-Status von Amazon EMR aus zu beschreiben. Dies ist erforderlich, damit Amazon EMR bestätigen kann, wann ein Cluster beendet wurde, und dann nach dreißig Tagen alle vom Cluster hinterlassenen WAL-Protokolle bereinigen kann.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeCluster"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS verwaltete Richtlinien für Amazon EMR

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. 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.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. 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 in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

Amazon EMR-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon EMR an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
AmazonEMRServicePolicy_v2 – Aktualisierung auf eine bestehende Richtlinie	Ab Amazon EMR Version 7.5.0 hinzugefügt <code>ec2:CreateVpcEndpoint</code> und für eine optimale Benutzere Erfahrung <code>ec2:CreateTags</code> erforderlich. <code>ec2:ModifyVpcEndpoint</code>	4. März 2025
AmazonEMRServicePolicy_v2 – Aktualisierung auf eine bestehende Richtlinie	Hinzugefügt <code>elasticmapreduce:CreatePersistentAppUI</code> , <code>elasticmapreduce:DescribePersistentAppUI</code> . <code>elasticmapreduce:GetPersistentAppUIPresignedURL</code>	28. Februar 2025

Änderung	Beschreibung	Datum
<u>EMRDescribeClusterPolicyForEMRWAL</u> – Neue Richtlinie	Es wurde eine neue Richtlinie hinzugefügt, sodass Amazon EMR dreißig Tage nach Beendigung des Clusters den Clusterstatus für die WAL-Bereinigung bestimmen kann.	10. August 2023
<u>AmazonEMRFullAccessPolicy_v2</u> und <u>AmazonEMRReadOnlyAccessPolicy_v2</u> – Zur Aktualisierung einer bestehenden Richtlinie	elasticmapreduce:DescribeReleaseLabel und elasticmapreduce:GetAutoTerminationPolicy hinzugefügt.	21. April 2022
<u>AmazonEMRFullAccessPolicy_v2</u> – Aktualisierung auf eine bestehende Richtlinie	ec2:DescribeImages hinzugefügt für <u>Verwendung eines benutzerdefinierten AMI für mehr Flexibilität bei der Amazon EMR-Clusterkonfiguration</u> .	15. Februar 2022
<u>Verwaltete Richtlinien von Amazon EMR</u>	Aktualisiert, um die Verwendung vordefinierter Benutzer-Tags zu verdeutlichen. Es wurde ein Abschnitt zur Verwendung der AWS Konsole zum Starten von Clustern mit verwalteten v2-Richtlinien hinzugefügt.	29. September 2021

Änderung	Beschreibung	Datum
<u>AmazonEMRFullAccessPolicy_v2</u> – Aktualisierung auf eine bestehende Richtlinie	Die Aktionen <code>PassRoleForAutoScaling</code> und <code>PassRoleForEC2</code> wurden dahingehend geändert, dass der <code>StringLike</code> -Bedingungsoperator entsprechend <code>"iam:PassedToService": "application-autoscaling.amazonaws.com"</code> und <code>"iam:PassedToService": "ec2.amazonaws.com"</code> verwendet wird.	20. Mai 2021
<u>AmazonEMRFullAccessPolicy_v2</u> – Aktualisierung auf eine bestehende Richtlinie	Ungültige Aktion <code>s3:ListBuckets</code> wurde entfernt und durch <code>s3:ListAllMyBuckets</code>-Aktion ersetzt. Die Erstellung von serviceverknüpften Rollen (SLR) wurde aktualisiert und ist nun explizit auf die einzige SLR beschränkt, die Amazon EMR mit expliziten Serviceprinzipien hat. Die SLRs, die erstellt werden können, sind genau die gleichen wie vor dieser Änderung.	23. März 2021

Änderung	Beschreibung	Datum
<u>AmazonEMRFullAccessPolicy_v2</u> – Neue Richtlinie	Amazon EMR hat neue Berechtigungen für den Bereichszugriff auf Ressourcen hinzugefügt und eine Voraussetzung hinzugefügt, dass Benutzer Ressourcen vordefinierte Benutzer-Tags hinzufügen müssen, bevor sie von Amazon EMR verwaltete Richtlinien verwenden können. <code>iam:PassRole</code> - Aktion erfordert, dass die <code>iam:PassedToService</code> - Bedingung auf den angegebenen Service gesetzt ist. Der Zugriff auf Amazon EC2, Amazon S3 und andere Dienste ist standardmäßig nicht erlaubt.	11. März 2021
<u>AmazonEMRServicePolicy_v2</u> – Neue Richtlinie	Fügt die Voraussetzung hinzu, dass Benutzer Benutzer-Tags zu Ressourcen hinzufügen müssen, bevor sie diese Richtlinie verwenden können.	11. März 2021
<u>AmazonEMRReadOnlyAccessPolicy_v2</u> – Neue Richtlinie	Mit Berechtigungen können nur bestimmte schreibgeschützte ElasticMapReduce-Aktionen ausgeführt werden. Der Zugriff auf Amazon S3 ist standardmäßig nicht zulässig.	11. März 2021

Änderung	Beschreibung	Datum
Amazon EMR hat mit der Nachverfolgung von Änderungen begonnen	Amazon EMR hat damit begonnen, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	11. März 2021

IAM-Richtlinien für Tag-basierten Zugriff auf Cluster und EMR-Notebooks

Sie können in Ihrer identitätsbasierten Richtlinie auf Tags basierende Bedingungen zum Steuern des Zugriffs auf Cluster und EMR-Notebooks verwenden.

Weitere Informationen zum Hinzufügen von Tags zu Clustern finden Sie unter [Tagging EMR Clusters](#).

Die folgenden Beispiele zeigen verschiedene Szenarien und Möglichkeiten zur Nutzung der Bedingungsoperatoren mit Amazon-EMR-Bedingungskontextschlüsseln. Diese IAM-Richtlinienanweisungen dienen nur zu Demonstrationszwecken und sollten nicht in Produktionsumgebungen verwendet werden. Es gibt mehrere Möglichkeiten für die Kombination von Richtlinienanweisungen zum Gewähren oder Verweigern von Berechtigungen entsprechend Ihren Anforderungen. Weitere Informationen zum Planen und Testen von IAM-Richtlinien finden Sie im [IAM-Benutzerhandbuch](#).

Important

Das explizite Ablehnen von Berechtigungen für Markierungsaktionen stellt eine wichtige Überlegung dar. Dadurch wird verhindert, dass Benutzer eine Ressource markieren und sich dadurch selbst Berechtigungen erteilen, die Sie nicht gewähren wollten. Wenn Sie Tagging-Aktionen für eine Ressource nicht verweigern, kann ein Benutzer Tags ändern und die Absicht der tagbasierten Richtlinien umgehen.

Beispiel identitätsbasierter Richtlinienanweisungen für Cluster

Die folgenden Beispiele zeigen identitätsbasierte Berechtigungsrichtlinien, die verwendet werden, um die Aktionen zu steuern, die mit EMR-Clustern zulässig sind.

Important

Für die `ModifyInstanceGroup`-Aktion in Amazon EMR müssen Sie keine Cluster-ID angeben. Aus diesem Grund sind zusätzliche Überlegungen erforderlich, um diese Aktion auf der Grundlage von Cluster-Tags abzulehnen. Weitere Informationen finden Sie unter [Die `ModifyInstanceGroup` Aktion in Amazon EMR ablehnen](#).

Themen

- [Zulassen von Aktionen nur für Cluster mit bestimmten Tag-Werten](#)
- [Erfordert Cluster-Tagging, wenn ein Cluster erstellt wird](#)
- [Aktionen für Cluster mit einem bestimmten Tag zulassen, unabhängig vom Tag-Wert](#)

Zulassen von Aktionen nur für Cluster mit bestimmten Tag-Werten

Die folgenden Beispiele veranschaulichen eine Richtlinie, mit der ein Benutzer Aktionen auf der Grundlage des Cluster-Tags `department` mit dem Wert `dev` durchführen, sowie Cluster mit demselben Tag markieren kann. Das letzte Richtlinienbeispiel zeigt, wie Sie Rechte zum Markieren von EMR-Clustern mit etwas anderem als demselben Tag ablehnen.

Im folgenden Richtlinienbeispiel versucht der `StringEquals`-Bedingungsoperator, `dev` und den Wert für das Tag `department` abzugleichen. Wenn das Tag `department` dem Cluster nicht hinzugefügt wurde oder den Wert `dev` nicht enthält, ist die Richtlinie nicht anzuwenden und die Aktionen werden von dieser Richtlinie nicht zugelassen. Wenn keine anderen Richtlinienanweisungen die Aktionen zulassen, kann der Benutzer nur mit Clustern arbeiten, die dieses Tag mit diesem Wert enthalten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt12345678901234",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:TerminateJobFlows",
        "elasticmapreduce:SetTerminationProtection",
        "elasticmapreduce:ListInstances",

```

```

    "elasticmapreduce:ListInstanceGroups",
    "elasticmapreduce:ListBootstrapActions",
    "elasticmapreduce:DescribeStep"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ResourceTag/department": "dev"
    }
  }
}
]
}

```

Sie können auch mehrere Tag-Werte mithilfe eines Bedingungsoperators angeben. Um beispielsweise alle Aktionen in Clustern zuzulassen, in denen das Tag *department* den Wert *dev* oder *test* enthält können Sie den Bedingungsblock im vorherigen Beispiel durch Folgendes ersetzen.

```

    "Condition": {
      "StringEquals": {
        "elasticmapreduce:ResourceTag/department":["dev", "test"]
      }
    }
  }
}

```

Erfordert Cluster-Tagging, wenn ein Cluster erstellt wird

Wie im oben stehenden Beispiel sucht die folgenden Beispielrichtlinie dasselbe übereinstimmende Tag: den Wert *dev* für das Tag *department*. In diesem Beispiel gibt der RequestTag-Bedingungsschlüssel jedoch an, dass die Richtlinie während der Tag-Erstellung gilt. Sie müssen also einen Cluster mit einem Tag erstellen, der dem angegebenen Wert entspricht.

Um einen Cluster mit einem Tag zu erstellen, benötigen Sie auch die Erlaubnis für die `elasticmapreduce:AddTags`-Aktion. Bei dieser Anweisung stellt der `elasticmapreduce:ResourceTag`-Bedingungsschlüssel sicher, dass IAM nur Zugriff auf Tag-Ressourcen gewährt, deren Wert *dev* auf dem *department*-Tag steht. Das Resource-Element wird verwendet, um diese Berechtigung auf Clusterressourcen zu beschränken.

Für die PassRole Ressourcen müssen Sie die AWS Konto-ID oder den Alias, den Namen der Servicerolle in der PassRoleForEMR Erklärung und den Namen des Instance-Profils in der PassRoleForEC2 Erklärung angeben. Weitere Informationen zum IAM-ARN-Format finden Sie unter [IAM ARNs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zum Abgleichen von Tag-Schlüsselwerten finden Sie unter [aws:RequestTag/tag-key](#) im IAM-Benutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RunJobFlowExplicitlyWithTag",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:RunJobFlow"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/department": "dev"
        }
      }
    },
    {
      "Sid": "AddTagsForDevClusters",
      "Effect": "Allow",
      "Action": "elasticmapreduce:AddTags",
      "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/department": "dev"
        }
      }
    },
    {
      "Sid": "PassRoleForEMR",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Sid": "PassRoleForEC2",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
    "Condition": {
      "StringLike": {
        "iam:PassedToService": "ec2.amazonaws.com*"
      }
    }
  }
]
}

```

Aktionen für Cluster mit einem bestimmten Tag zulassen, unabhängig vom Tag-Wert

Sie können auch Aktionen nur für Cluster mit einem bestimmten Tag, unabhängig vom Tag-Wert, zulassen. Dazu können Sie den `Null`-Operator verwenden. Weitere Informationen finden Sie unter [Bedingungsoperator zum Überprüfen der Existenz von Bedingungsschlüsseln](#) im IAM-Benutzerhandbuch. Um beispielsweise Aktionen nur in EMR-Clustern mit dem Tag *department* unabhängig von dem enthaltenen Wert, zuzulassen, können Sie die Bedingungsblöcke im vorherigen Beispiel durch Folgendes ersetzen. Der `Null`-Operator sucht einem vorhandenen *department*-Tag in einem EMR-Cluster. Wenn das Tag vorhanden ist, wird die Anweisung `Null` entsprechend der in dieser Richtlinienanweisung angegebenen Bedingung mit `"false"` ausgewertet und die jeweiligen Aktionen werden zugelassen.

```

"Condition": {
  "Null": {
    "elasticmapreduce:ResourceTag/department": "false"
  }
}

```

Mit der folgenden Richtlinienanweisung kann ein Benutzer einen EMR-Cluster nur erstellen, wenn der Cluster über ein *department*-Tag mit einem beliebigen Wert verfügt. Für die `PassRole` Ressource müssen Sie die AWS Konto-ID oder den Alias und den Namen der Servicerolle angeben. Weitere Informationen zum IAM-ARN-Format finden Sie unter [IAM ARNs](#) im IAM-Benutzerhandbuch.

Weitere Informationen finden Sie unter [Bedingungsoperator zum Überprüfen der Existenz von Bedingungsschlüsseln](#) im IAM-Benutzerhandbuch.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateClusterTagNullCondition",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:RunJobFlow"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/department": "false"
        }
      }
    },
    {
      "Sid": "AddTagsNullCondition",
      "Effect": "Allow",
      "Action": "elasticmapreduce:AddTags",
      "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
      "Condition": {
        "Null": {
          "elasticmapreduce:ResourceTag/department": "false"
        }
      }
    },
    {
      "Sid": "PassRoleForElasticMapReduce",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
        }
      }
    }
  ],
}
```

```

    {
      "Sid": "PassRoleForEC2",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "ec2.amazonaws.com*"
        }
      }
    }
  ]
}

```

Beispielhafte identitätsbasierte Richtlinienanweisungen für EMR Notebooks

Die IAM-Beispiel-Richtlinienanweisungen in diesem Abschnitt zeigen häufige Szenarien für die Verwendung von Schlüsseln, um zulässige Aktionen mit EMR Notebooks zu beschränken. Solange keine andere mit dem Prinzipal (Benutzer) verknüpfte Richtlinie die Aktionen zulässt, schränken die Bedingungskontextschlüssel die zulässigen Aktionen wie angegeben ein.

Example – Erlaubt nur den Zugriff auf EMR Notebooks, die ein Benutzer auf der Grundlage von Tagging erstellt

Wenn die folgende Beispiel-Richtlinienanweisung an eine Rolle oder einen Benutzer angefügt wird, können Benutzer nur mit Notebooks arbeiten, die sie selbst erstellt haben. Diese Richtlinienanweisung verwendet das bei der Erstellung eines Notebooks angewendete Standard-Tag.

In diesem Beispiel versucht der Bedingungsoperator `StringEquals`, eine Variable, die die Benutzer-ID (`{aws:userId}`) des aktuellen Benutzers darstellt, dem Wert des Tags `creatorUserID` zuzuordnen. Wenn das Tag `creatorUserID` nicht zum Notebook hinzugefügt wurde oder den Wert der ID des aktuellen Benutzers nicht enthält, ist die Richtlinie nicht anzuwenden und die Aktionen werden von dieser Richtlinie nicht zugelassen. Wenn keine anderen Richtlinienanweisungen die Aktionen zulassen, kann der Benutzer nur mit Notebooks arbeiten, die dieses Tag mit diesem Wert enthalten.

```

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

```

```

        "elasticmapreduce:DescribeEditor",
        "elasticmapreduce:StartEditor",
        "elasticmapreduce:StopEditor",
        "elasticmapreduce>DeleteEditor",
        "elasticmapreduce:OpenEditorInConsole"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
        }
    }
}
]
}

```

Example – Notebook-Tagging anfordern, wenn ein Notebook erstellt wird

In diesem Beispiel wird der Kontextschlüssel `RequestTag` verwendet. Die Aktion `CreateEditor` ist nur dann zulässig, wenn der Benutzer das `creatorUserId` Tag, das standardmäßig hinzugefügt wird, nicht ändert oder löscht. Die Variable `${aws:userId}` gibt die Benutzer-ID des aktuell aktiven Benutzers an. Dies ist der Standardwert des Tags.

Die Richtlinienanweisung kann verwendet werden, um sicherzustellen, dass Benutzer das Tag `createUserId` nicht entfernen und dessen Wert nicht ändern.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:CreateEditor"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
            "elasticmapreduce:RequestTag/creatorUserId": "${aws:userid}"
          }
      }
    }
  ]
}

```

```
}
```

Dieses Beispiel erfordert, dass der Benutzer den Cluster mit einem Tag mit der Schlüsselzeichenfolge `dept` und einem der folgenden Werte erstellt: `datascience`, `analytics`, `operations`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:CreateEditor"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:RequestTag/dept": [
            "datascience",
            "analytics",
            "operations"
          ]
        }
      }
    }
  ]
}
```

Example – Die Notebook-Erstellung auf getaggte Cluster beschränken und Notebook-Tags anfordern

Dieses Beispiel erlaubt die Notebook-Erstellung nur, wenn das Notebook mit einem Tag erstellt wird, bei dem die Schlüsselzeichenfolge `owner` auf einen der angegebenen Werte festgelegt ist. Darüber hinaus kann das Notebook nur erstellt werden, wenn der Cluster ein Tag enthält, bei dem die Schlüsselzeichenfolge `department` auf einen der angegebenen Werte festgelegt ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:CreateEditor"
      ],
```

```

    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:RequestTag/owner": [
                "owner1",
                "owner2",
                "owner3"
            ],
            "elasticmapreduce:ResourceTag/department": [
                "dep1",
                "dep3"
            ]
        }
    }
}

```

Example – Basierend auf Tags die Möglichkeit einschränken, ein Notebook zu starten

Dieses Beispiel schränkt die Möglichkeit, ein Notebook zu starten, auf Notebooks ein, die ein Tag enthalten, bei dem die Schlüsselzeichenfolge `owner` auf einen der angegebenen Werte festgelegt ist. Da das Element `Resource` verwendet wird, um nur den `editor` anzugeben, gilt die Bedingung nicht für den Cluster und ein Tagging ist nicht erforderlich.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:StartEditor"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/owner": [
            "owner1",
            "owner2"
          ]
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

Dieses Beispiel ähnelt dem obigen. Die Einschränkung gilt hier jedoch nur für getaggte Cluster, nicht für Notebooks.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:StartEditor"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/department": [
            "dep1",
            "dep3"
          ]
        }
      }
    }
  ]
}

```

Dieses Beispiel verwendet andere Notebook- und Cluster-Tags. Es ermöglicht das Starten eines Notebooks nur, wenn Folgendes zutrifft:

- Das Notebook enthält ein Tag, bei dem die Schlüsselzeichenfolge `owner` auf einen der angegebenen Wert festgelegt ist.
- und –
- Der Cluster enthält ein Tag, bei dem die Schlüsselzeichenfolge `department` auf einen der angegebenen Wert festgelegt ist.

```

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

```



```

{
  "Action": [
    "elasticmapreduce:StartEditor"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ResourceTag/owner": [
        "user1",
        "user2"
      ]
    }
  }
},
{
  "Action": [
    "elasticmapreduce:StartEditor"
  ],
  "Effect": "Allow",
  "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
  "Condition": {
    "StringEquals": {
      "elasticmapreduce:ResourceTag/department": [
        "datascience",
        "analytics"
      ]
    }
  }
}
]
}

```

Example – Basierend auf Tags die Möglichkeit einschränken, den Notebook-Editor zu öffnen

In diesem Beispiel kann der Notebook-Editor nur geöffnet werden, wenn Folgendes zutrifft:

- Das Notebook enthält ein Tag, bei dem die Schlüsselzeichenfolge `owner` auf einen der angegebenen Wert festgelegt ist.
- und –
- Der Cluster enthält ein Tag, bei dem die Schlüsselzeichenfolge `department` auf einen der angegebenen Wert festgelegt ist.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:OpenEditorInConsole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:elasticmapreduce*:123456789012:editor/*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/owner": [
            "user1",
            "user2"
          ]
        }
      }
    },
    {
      "Action": [
        "elasticmapreduce:OpenEditorInConsole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:elasticmapreduce*:123456789012:cluster/*",
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/department": [
            "datascience",
            "analytics"
          ]
        }
      }
    }
  ]
}
```

Die ModifyInstanceGroup Aktion in Amazon EMR ablehnen

Für die [ModifyInstanceGroups](#) Aktion in Amazon EMR müssen Sie bei der Aktion keine Cluster-ID angeben. Stattdessen können Sie nur eine Instance-Gruppen-ID angeben. Aus diesem Grund hat eine scheinbar einfache Ablehnungsrichtlinie für diese Aktion, die auf der Cluster-ID oder einem

Cluster-Tag basiert, möglicherweise nicht die beabsichtigte Wirkung. Betrachten Sie die folgende Beispielrichtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:ModifyInstanceGroups"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "elasticmapreduce:ModifyInstanceGroups"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/j-12345ABCDEF67"
    }
  ]
}
```

Wenn ein Benutzer, dem diese Richtlinie zugewiesen ist, eine `ModifyInstanceGroup`-Aktion ausführt und nur die Instance-Gruppen-ID angibt, gilt die Richtlinie nicht. Da die Aktion für alle anderen Ressourcen zulässig ist, ist die Aktion erfolgreich.

Eine Lösung für dieses Problem besteht darin, der Identität eine Richtlinienerklärung beizufügen, die ein [NotResource](#) Element verwendet, um jede `ModifyInstanceGroup` Aktion abzulehnen, die ohne Cluster-ID ausgeführt wird. Die folgende Beispielrichtlinie fügt eine solche Deny-Anweisung hinzu, sodass jede `ModifyInstanceGroups`-Anfrage fehlschlägt, sofern keine Cluster-ID angegeben ist. Da eine Identität bei der Aktion eine Cluster-ID angeben muss, sind Ablehnungsbefehle, die auf der Cluster-ID basieren, daher wirksam.

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

```

        "elasticmapreduce:ModifyInstanceGroups"
    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "elasticmapreduce:ModifyInstanceGroups"
    ],
    "Effect": "Deny",
    "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEF67"
  },
  {
    "Action": [
      "elasticmapreduce:ModifyInstanceGroups"
    ],
    "Effect": "Deny",
    "NotResource": "arn:*:elasticmapreduce:*:*:cluster/*"
  }
]
}

```

Ein ähnliches Problem tritt auf, wenn Sie die `ModifyInstanceGroups`-Aktion auf der Grundlage des mit einem Cluster-Tag verknüpften Werts ablehnen möchten. Die Lösung ist ähnlich. Zusätzlich zu einer Ablehnungs-Anweisung, die den Tag-Wert angibt, können Sie eine Richtlinienanweisung hinzufügen, die die `ModifyInstanceGroup`-Aktion ablehnt, wenn das von Ihnen angegebene Tag nicht vorhanden ist, unabhängig vom Wert.

Das folgende Beispiel zeigt eine Richtlinie, die, wenn sie an eine Identität angehängt ist, der Identität die `ModifyInstanceGroups`-Aktion aller Cluster verweigert, bei denen das Tag `department` auf `dev` gesetzt ist. Diese Anweisung ist nur aufgrund der Ablehnungs-Anweisung wirksam, die die `StringNotLike`-Bedingung verwendet, um die Aktion zu verweigern, sofern das `department`-Tag nicht vorhanden ist.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:ModifyInstanceGroups"

```

```

    ],
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Action": [
      "elasticmapreduce:ModifyInstanceGroups"
    ],
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/department": "dev"
      }
    },
    "Effect": "Deny",
    "Resource": "*"
  },
  {
    "Action": [
      "elasticmapreduce:ModifyInstanceGroups"
    ],
    "Condition": {
      "StringNotLike": {
        "aws:ResourceTag/department": "?*"
      }
    },
    "Effect": "Deny",
    "Resource": "*"
  }
],
}

```

Fehlerbehebung für Amazon-EMR-Identität und -Zugriff

Diagnostizieren und beheben Sie mithilfe der folgenden Informationen gängige Probleme, die bei der Verwendung von Amazon EMR und IAM auftreten können.

Themen

- [Ich bin nicht autorisiert, eine Aktion in Amazon EMR auszuführen](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amazon EMR-Ressourcen ermöglichen](#)

Ich bin nicht autorisiert, eine Aktion in Amazon EMR auszuführen

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion auszuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven `my-example-widget`-Ressource zu verwenden, jedoch nicht über EMR: `GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
EMR: GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `my-example-widget` auf die Ressource EMR: `GetWidget` zugreifen zu können.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der Aktion `iam:PassRole` autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon EMR übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon EMR auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:  
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amazon EMR-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon EMR diese Features unterstützt, finden Sie unter [Funktionsweise von Amazon EMR mit IAM](#).
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen in AWS-Konten Ihrem Besitz gewähren können, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto, dem Sie](#) gehören.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Verwenden von Amazon S3 Access Grants mit Amazon EMR

Übersicht über S3 Access Grants für Amazon EMR

Mit den Amazon-EMR-Versionen 6.15.0 und höher bieten Amazon S3 Access Grants eine skalierbare Zugriffskontrolllösung, mit der Sie den Zugriff auf Ihre Amazon-S3-Daten von Amazon EMR aus erweitern können. Wenn Sie für Ihre S3-Daten eine komplexe oder umfangreiche Berechtigungskonfiguration haben, können Sie mit S3 Access Grants die S3-Datenberechtigungen für Benutzer, Gruppen, Rollen und Anwendungen in Ihrem Cluster skalieren.

Verwenden Sie S3 Access Grants, um den Zugriff auf Amazon-S3-Daten über die Berechtigungen hinaus zu erweitern, die durch die Laufzeitrolle oder die IAM-Rollen gewährt werden, die den

Identitäten mit Zugriff auf Ihren EMR-Cluster zugewiesen sind. Weitere Informationen finden Sie unter [Verwalten des Zugriffs mit S3 Access Grants](#) im Benutzerhandbuch zu Amazon S3.

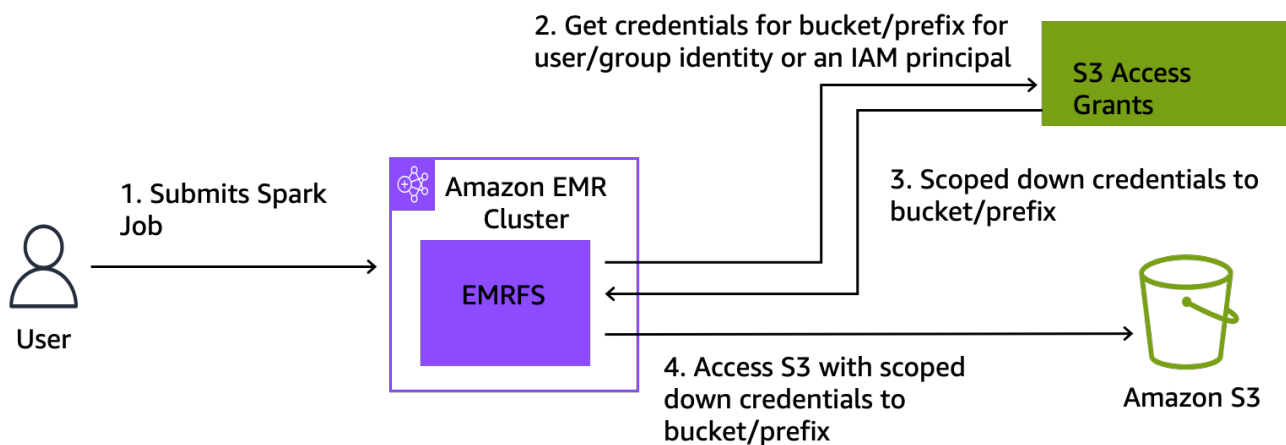
Schritte zur Verwendung von S3 Access Grants mit anderen Amazon-EMR-Bereitstellungen finden Sie in der folgenden Dokumentation:

- [Verwenden von S3 Access Grants mit Amazon EMR in EKS](#)
- [Verwenden von S3 Access Grants mit Amazon EMR Serverless](#)

So funktioniert Amazon EMR mit S3 Access Grants

Amazon-EMR-Versionen 6.15.0 und höher bieten eine native Integration mit S3 Access Grants. Sie können S3 Access Grants in Amazon EMR aktivieren und Spark-Aufträge ausführen. Wenn ein Spark-Auftrag eine Anfrage für S3-Daten stellt, stellt Amazon S3 temporäre Anmeldeinformationen bereit, die auf den jeweiligen Bucket, das Präfix oder das Objekt beschränkt sind.

Nachfolgend finden Sie einen allgemeinen Überblick darüber, wie Amazon EMR Zugriff auf Daten erhält, die durch S3 Access Grants geschützt sind.



1. Ein Benutzer reicht einen Amazon-EMR-Spark-Auftrag ein, der in Amazon S3 gespeicherte Daten verwendet.
2. Amazon EMR stellt eine Anfrage für S3 Access Grants, um im Namen dieses Benutzers Zugriff auf den Bucket, das Präfix oder das Objekt zu gewähren.
3. Amazon S3 gibt temporäre Anmeldeinformationen in Form eines AWS Security Token Service (STS-) Tokens für den Benutzer zurück. Das Token ist für den Zugriff auf den S3-Bucket, das S3-Präfix oder das S3-Objekt vorgesehen.
4. Amazon EMR verwendet das STS-Token, um Daten aus S3 abzurufen.

5. Amazon EMR empfängt die Daten von S3 und gibt die Ergebnisse an den Benutzer zurück.

Überlegungen zu S3 Access Grants mit Amazon EMR

Beachten Sie die folgenden Verhaltensweisen und Einschränkungen, wenn Sie S3 Access Grants mit Amazon EMR verwenden.

Feature-Unterstützung

- S3 Access Grants wird mit den Amazon-EMR-Versionen 6.15.0 und höher unterstützt.
- Spark ist die einzige unterstützte Abfrage-Engine, wenn Sie S3 Access Grants mit Amazon EMR verwenden.
- Delta Lake und Hudi sind die einzigen unterstützten Open-Table-Formate, wenn Sie S3 Access Grants mit Amazon EMR verwenden.
- Die folgenden Amazon-EMR-Funktionen werden bei der Verwendung mit S3 Access Grants nicht unterstützt:
 - Apache-Iceberg-Tabellen
 - Native LDAP-Authentifizierung
 - Native Apache-Ranger-Authentifizierung
 - AWS CLI Anfragen an Amazon S3, die IAM-Rollen verwenden
 - S3-Zugriff über das Open-Source-Protokoll S3A
- Die Option `fallbackToIAM` wird nicht für EMR-Cluster unterstützt, die die Verbreitung vertrauenswürdiger Identitäten mit IAM Identity Center verwenden.
- [S3 Access Grants mit AWS Lake Formation](#) wird nur mit Amazon EMR-Clustern unterstützt, die auf Amazon EC2 ausgeführt werden.

Überlegungen in Bezug auf das Verhalten

- Die native Apache-Ranger-Integration in Amazon EMR bietet Funktionen, die S3 Access Grants als Teil des Plugins EMRFS S3 Apache Ranger entsprechen. Wenn Sie Apache Ranger für differenzierte Zugriffskontrolle (FGAC) verwenden, empfehlen wir, dieses Plugin anstelle von S3 Access Grants zu verwenden.
- Amazon EMR stellt einen Cache für Anmeldeinformationen in EMRFS bereit, um sicherzustellen, dass ein Benutzer innerhalb eines Spark-Auftrags nicht wiederholt dieselben

Anmeldeinformationen anfordern muss. Daher fordert Amazon EMR bei der Anforderung von Anmeldeinformationen immer die Standardberechtigung an. Weitere Informationen finden Sie unter [Zugriff auf S3-Daten anfordern](#) im Benutzerhandbuch zu Amazon S3.

- Für den Fall, dass ein Benutzer eine Aktion ausführt, die S3 Access Grants nicht unterstützt, ist Amazon EMR so eingestellt, dass es die IAM-Rolle verwendet, die für die Auftragsausführung angegeben wurde. Weitere Informationen finden Sie unter [Fallback auf IAM-Rollen](#).

Starten eines Amazon-EMR-Clusters mit S3 Access Grants

In diesem Abschnitt wird beschrieben, wie Sie einen EMR-Cluster starten EC2, der auf Amazon läuft und S3 Access Grants verwendet, um den Zugriff auf Daten in Amazon S3 zu verwalten. Schritte zur Verwendung von S3 Access Grants mit anderen Amazon-EMR-Bereitstellungen finden Sie in der folgenden Dokumentation:

- [Verwenden von S3 Access Grants mit Amazon EMR in EKS](#)
- [Verwenden von S3 Access Grants mit EMR Serverless](#)

Gehen Sie wie folgt vor, um einen EMR-Cluster zu starten EC2, der auf Amazon läuft und S3 Access Grants verwendet, um den Zugriff auf Daten in Amazon S3 zu verwalten.

1. Richten Sie eine Auftragsausführungsrolle für Ihren EMR-Cluster ein. Geben Sie die erforderlichen IAM-Berechtigungen an, die Sie für die Ausführung von Spark-Aufträgen benötigen, `s3:GetDataAccess` und `s3:GetAccessGrantsInstanceForPrefix`:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:GetDataAccess",
    "s3:GetAccessGrantsInstanceForPrefix"
  ],
  "Resource": [
    //LIST ALL INSTANCE ARNS THAT THE ROLE IS ALLOWED TO QUERY
    "arn:aws_partition:s3:Region:account-id1:access-grants/default",
    "arn:aws_partition:s3:Region:account-id2:access-grants/default"
  ]
}
```

Note

Mit Amazon EMR erweitern S3 Access Grants die Berechtigungen, die in IAM-Rollen festgelegt sind. Wenn die IAM-Rollen, die Sie für die Auftragsausführung angeben, Berechtigungen für den direkten Zugriff auf S3 enthalten, können Benutzer möglicherweise auf mehr Daten zugreifen als nur auf die Daten, die Sie in S3 Access Grants definieren.

2. Verwenden Sie als Nächstes die, AWS CLI um einen Cluster mit Amazon EMR 6.15 oder höher und die `emrfs-site` Klassifizierung zu erstellen, um S3 Access Grants zu aktivieren, ähnlich dem folgenden Beispiel:

```
aws emr create-cluster
  --release-label emr-6.15.0 \
  --instance-count 3 \
  --instance-type m5.xlarge \
  --configurations '[{"Classification":"emrfs-site",
"Properties":{"fs.s3.s3AccessGrants.enabled":"true",
"fs.s3.s3AccessGrants.fallbackToIAM":"false"}}]'
```

S3 Access Grants mit AWS Lake Formation

Wenn Sie Amazon EMR mit der [AWS Lake Formation -Integration](#) verwenden, können Sie Amazon S3 Access Grants für direkten oder tabellarischen Zugriff auf Daten in Amazon S3 verwenden.

Note

S3 Access Grants mit AWS Lake Formation wird nur mit Amazon EMR-Clustern unterstützt, die auf Amazon EC2 ausgeführt werden.

Direkter Zugriff

Der direkte Zugriff umfasst alle Aufrufe zum Zugriff auf S3-Daten, die nicht die API für den AWS Glue-Service aufrufen, den Lake Formation als Metastore mit Amazon EMR verwendet, zum Beispiel, um Folgendes aufzurufen: `spark.read`

```
spark.read.csv("s3://...")
```

Wenn Sie S3 Access Grants AWS Lake Formation auf Amazon EMR verwenden, durchlaufen alle Direktzugriffsmuster S3 Access Grants, um temporäre S3-Anmeldeinformationen zu erhalten.

Tabellarischer Zugriff

Ein tabellarischer Zugriff erfolgt, wenn Lake Formation die Metastore-API aufruft, um auf Ihren S3-Standort zuzugreifen, z. B. um Tabellendaten abzufragen:

```
spark.sql("select * from test_tbl")
```

Wenn Sie S3 Access Grants mit AWS Lake Formation auf Amazon EMR verwenden, durchlaufen alle tabellarischen Zugriffsmuster Lake Formation.

Fallback auf IAM-Rollen

Wenn ein Benutzer versucht, eine Aktion auszuführen, die S3 Access Grants nicht unterstützt, wird Amazon EMR standardmäßig die IAM-Rolle verwenden, die für die Auftragsausführung angegeben wurde, wenn die `fallbackToIAM`-Konfiguration `true` ist. Auf diese Weise können Benutzer in Szenarien, die S3 Access Grants nicht abdeckt, auf ihre Auftragsausführungsrolle zurückgreifen, um Anmeldeinformationen für den S3-Zugriff einzugeben.

Wenn `fallbackToIAM` aktiviert ist, können Benutzer auf die Daten zugreifen, die Access Grant zulässt. Wenn es kein S3-Access-Grants-Token für die Zieldaten gibt, überprüft Amazon EMR, ob die entsprechende Berechtigung für die Auftragsausführungsrolle vorliegt.

Note

Wir empfehlen Ihnen, Ihre Zugriffsberechtigungen bei aktivierter `fallbackToIAM`-Konfiguration zu testen, auch wenn Sie planen, die Option für Produktionsworkloads zu deaktivieren. Bei Spark-Aufträgen gibt es andere Möglichkeiten, wie Benutzer mit ihren IAM-Anmeldeinformationen auf alle Berechtigungssätze zugreifen können. Wenn sie auf EMR-Clustern aktiviert sind, gewähren S3-Erteilungen Spark-Aufträgen Zugriff auf S3-Standorte. Sie sollten sicherstellen, dass Sie diese S3-Standorte vor Zugriffen außerhalb von EMRFS schützen. Sie sollten die S3-Standorte beispielsweise vor dem Zugriff durch S3-Clients schützen, die in Notebooks verwendet werden, oder durch Anwendungen, die nicht von S3 Access Grants unterstützt werden, wie Hive oder Presto.

Authentifizieren von Amazon-EMR-Cluster-Knoten

SSH-Clients können ein EC2 Amazon-Schlüsselpaar verwenden, um sich bei Cluster-Instances zu authentifizieren. Alternativ können Sie bei Amazon EMR Version 5.10.0 oder höher Kerberos so konfigurieren, dass Benutzer und SSH-Verbindungen gegenüber dem Primärknoten authentifiziert werden. Und mit den Amazon-EMR-Versionen 5.12.0 und höher können Sie sich mit LDAP authentifizieren.

Themen

- [Verwenden Sie ein EC2 key pair für SSH-Anmeldeinformationen für Amazon EMR](#)
- [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#)
- [Active-Directory- oder LDAP-Server für die Authentifizierung mit Amazon EMR verwenden](#)

Verwenden Sie ein EC2 key pair für SSH-Anmeldeinformationen für Amazon EMR

Amazon EMR-Clusterknoten werden auf EC2 Amazon-Instances ausgeführt. Sie können eine Verbindung zu Clusterknoten genauso herstellen wie zu EC2 Amazon-Instances. Sie können Amazon verwenden EC2 , um ein key pair zu erstellen, oder Sie können ein key pair importieren. Wenn Sie einen Cluster erstellen, können Sie das EC2 Amazon-Schlüsselpaar angeben, das für SSH-Verbindungen zu allen Cluster-Instances verwendet wird. Außerdem können Sie auch einen Cluster ohne ein Schlüsselpaar erstellen. Dies geschieht normalerweise mit vorübergehenden Clusters, die starten, gewisse Schritte ausführen und dann automatisch beendet werden.

Der SSH-Client, mit dem Sie sich mit dem Cluster verbinden, benötigt die Datei mit dem privaten Schlüssel, der mit diesem Schlüsselpaar verknüpft ist. Dies ist eine .pem-Datei für SSH-Clients unter Linux, Unix und macOS. Sie müssen die Berechtigungen so festlegen, dass nur der Schlüsselbesitzer berechtigt ist, auf die Datei zuzugreifen. Dies ist eine .ppk-Datei für SSH-Clients mit Windows, und die .ppk-Datei wird in der Regel von der .pem-Datei erstellt.

- Weitere Informationen zum Erstellen eines EC2 Amazon-Schlüsselpaars finden Sie unter [EC2 Amazon-Schlüsselpaare](#) im EC2 Amazon-Benutzerhandbuch.
- Anweisungen zur Verwendung von PuTTY zum Erstellen einer .ppk-Datei aus einer .pem-Datei finden Sie unter [Konvertieren Ihres privaten Schlüssels mit PuTTY im Amazon-Benutzerhandbuch](#). EC2

- Weitere Informationen zum Festlegen von Zugriffsberechtigungen für PEM-Dateien und zum Herstellen einer Verbindung mit dem primären Knoten eines EMR-Clusters mithilfe verschiedener Methoden — einschließlich ssh Linux oder macOS, PuTTY unter Windows oder von einem beliebigen unterstützten Betriebssystem AWS CLI aus — finden Sie unter. [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#)

Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR

Amazon-EMR-Versionen 5.10.0 und höher unterstützen Kerberos. Kerberos ist ein Netzwerkauthentifizierungsprotokoll, das eine Verschlüsselung mit geheimen Schlüsseln verwendet, um eine starke Authentifizierung bereitzustellen, sodass Passwörter oder andere Anmeldeinformationen nicht in einem unverschlüsselten Format über das Netzwerk gesendet werden.

In Kerberos werden Services und Benutzer, die sich authentifizieren müssen, als Prinzipale bezeichnet. Prinzipale befinden sich in einem Kerberos-Bereich. Innerhalb des Bereichs gewährt ein Kerberos-Server, der als Key Distribution Center (KDC) bezeichnet wird, Prinzipalen die Möglichkeit, sich zu authentifizieren. Dazu stellt das KDC Tickets für die Authentifizierung aus. Das KDC unterhält eine Datenbank der Prinzipale in seinem Bereich mit ihren Passwörtern und anderen administrativen Informationen zu jedem Prinzipal. Ein KDC kann auch Anmeldeinformationen für die Authentifizierung von Prinzipalen aus anderen Bereichen akzeptieren. Dies wird als bereichsübergreifendes Vertrauen bezeichnet. Darüber hinaus kann ein EMR-Cluster ein externes KDC verwenden, um Prinzipale zu authentifizieren.

Ein gängiges Szenario für die Einrichtung einer bereichsübergreifenden Vertrauensstellung oder für die Verwendung eines externen KDC ist die Authentifizierung von Benutzern von einer Active-Directory-Domain aus. Auf diese Weise können Benutzer mit ihrem Domainkonto auf einen EMR-Cluster zugreifen, wenn sie SSH verwenden, um eine Verbindung zu einem Cluster herzustellen oder mit Big-Data-Anwendungen zu arbeiten.

Bei Verwendung der Kerberos-Authentifizierung konfiguriert Amazon EMR Kerberos für die Anwendungen, Komponenten und Subsysteme, die es auf dem Cluster installiert, damit sie gegenseitig authentifiziert werden.

Important

Amazon EMR unterstützt nicht AWS Directory Service for Microsoft Active Directory in einem realmübergreifenden Trust oder als externes KDC.

Bevor Sie Kerberos mit Amazon EMR konfigurieren, empfehlen wir, dass Sie sich über Kerberos-Konzepte, die Services für die Ausführung auf einem KDC und die Tools für die Verwaltung von Kerberos-Services informieren. Weitere Informationen finden Sie in der [MIT-Kerberos-Dokumentation](#), die vom [Kerberos Consortium](#) veröffentlicht wird.

Themen

- [Unterstützte Anwendungen mit Amazon EMR](#)
- [Kerberos-Architekturoptionen mit Amazon EMR](#)
- [Konfiguration von Kerberos in Amazon EMR](#)
- [Verwenden von SSH zum Herstellen einer Verbindung zu kerberisierten Clustern mit Amazon EMR](#)
- [Tutorial: Konfiguration eines clusterspezifischen KDC mit Amazon EMR](#)
- [Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain](#)

Unterstützte Anwendungen mit Amazon EMR

In einem EMR-Cluster sind Kerberos-Prinzipale die Big Data-Anwendungsservices und Sub-Systemen, die auf allen Cluster-Knoten ausgeführt werden. Amazon EMR kann die Anwendungen und die unten aufgeführten Komponenten für die Verwendung von Kerberos konfigurieren. Jeder Anwendung ist ein Kerberos-Benutzer-Prinzipal zugeordnet.

Amazon EMR unterstützt kein bereichsübergreifendes Vertrauen für AWS Directory Service for Microsoft Active Directory.

Amazon EMR konfiguriert nur die Open-Source-Authentifizierungsfeatures für Kerberos für die unten aufgelisteten Anwendungen und Komponenten. Alle anderen installierten Anwendungen sind nicht durch Kerberos geschützt. Dies kann zu einer Unfähigkeit der Kommunikation mit durch Kerberos geschützten Komponenten führen und Anwendungsfehler verursachen. Für Anwendungen und Komponenten, die nicht durch Kerberos geschützt sind, ist keine Authentifizierung aktiviert. Die unterstützten Anwendungen und Komponenten können je nach Amazon EMR Versionen variieren.

Die Livy-Benutzeroberfläche ist die einzige Weboberfläche, die auf dem Kerberized Cluster gehostet wird.

- Hadoop MapReduce
- Hbase
- HCatalog

- HDFS
- Hive
 - Aktivieren Sie keine Hive mit LDAP-Authentifizierung. Dies kann Probleme bei der Kommunikation mit durch Kerberos geschütztem YARN verursachen.
- Hue
 - Die Hue-Benutzerauthentifizierung wird nicht automatisch eingerichtet und kann unter Verwendung der Konfigurations-API konfiguriert werden.
 - Der Hue-Server ist durch Kerberos geschützt. Das Hue Front-End (UI) ist nicht für die Authentifizierung konfiguriert. Die LDAP-Authentifizierung kann für die Hue-UI konfiguriert werden.
- Livy
 - Livy-Identitätswechsel mit kerberisierten Clustern wird in Amazon-EMR-Versionen 5.22.0 und höher unterstützt.
- Oozie
- Phoenix
- Presto
 - Presto unterstützt die Kerberos-Authentifizierung in Amazon-EMR-Versionen 6.9.0 und höher.
 - [Um die Kerberos-Authentifizierung für Presto zu verwenden, müssen Sie die Verschlüsselung bei der Übertragung aktivieren.](#)
- Spark
- Tez
- Trino
 - Trino unterstützt die Kerberos-Authentifizierung in Amazon-EMR-Versionen 6.11.0 und höher.
 - [Um die Kerberos-Authentifizierung für Trino zu verwenden, müssen Sie die Verschlüsselung bei der Übertragung aktivieren.](#)
- YARN
- Zeppelin
 - Zeppelin ist nur mit dem Spark-Interpreter für die Verwendung von Kerberos konfiguriert. Es ist nicht für andere Interpreter konfiguriert.
 - Der Identitätswechsel von Benutzern wird für Kerberized-Zeppelin-Interpreter außer Spark nicht unterstützt.
- Zookeeper

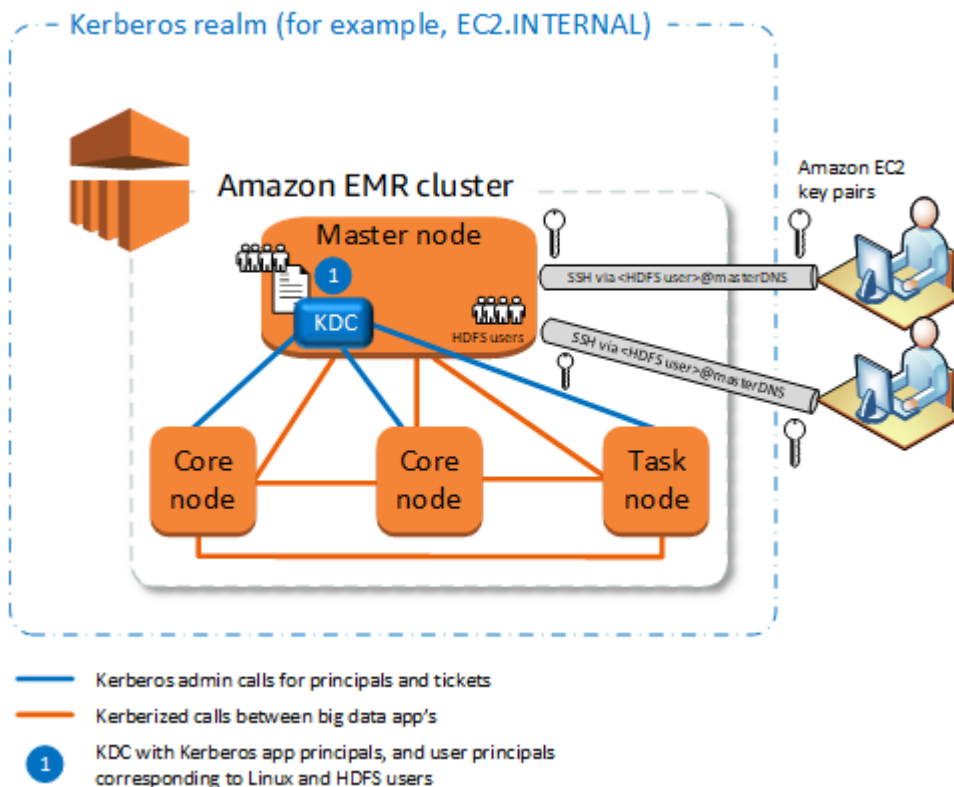
- Der Zookeeper-Client wird nicht unterstützt.

Kerberos-Architekturoptionen mit Amazon EMR

Wenn Sie Kerberos mit Amazon EMR verwenden, können Sie aus den in diesem Abschnitt aufgeführten Architekturen wählen. Unabhängig von der gewählten Architektur konfigurieren Sie Kerberos anhand der gleichen Schritte. Sie erstellen eine Sicherheitskonfiguration, legen die Sicherheitskonfiguration und kompatible Cluster-spezifische Kerberos-Optionen fest, wenn Sie den Cluster erstellen. Zudem erstellen Sie HDFS-Verzeichnisse für Linux-Benutzer auf dem Cluster, der den Benutzerprinzipalen auf dem KDC entspricht. Weitere Informationen zu Konfigurationsoptionen und Beispielkonfigurationen für jede Architektur finden Sie unter [Konfiguration von Kerberos in Amazon EMR](#).

Cluster-spezifisches KDC (KDC auf dem Primärknoten)

Diese Konfiguration ist nur mit Amazon-EMR-Versionen 5.10.0 und höher verfügbar.



Vorteile

- Amazon EMR ist im vollständigen Besitz des KDC.

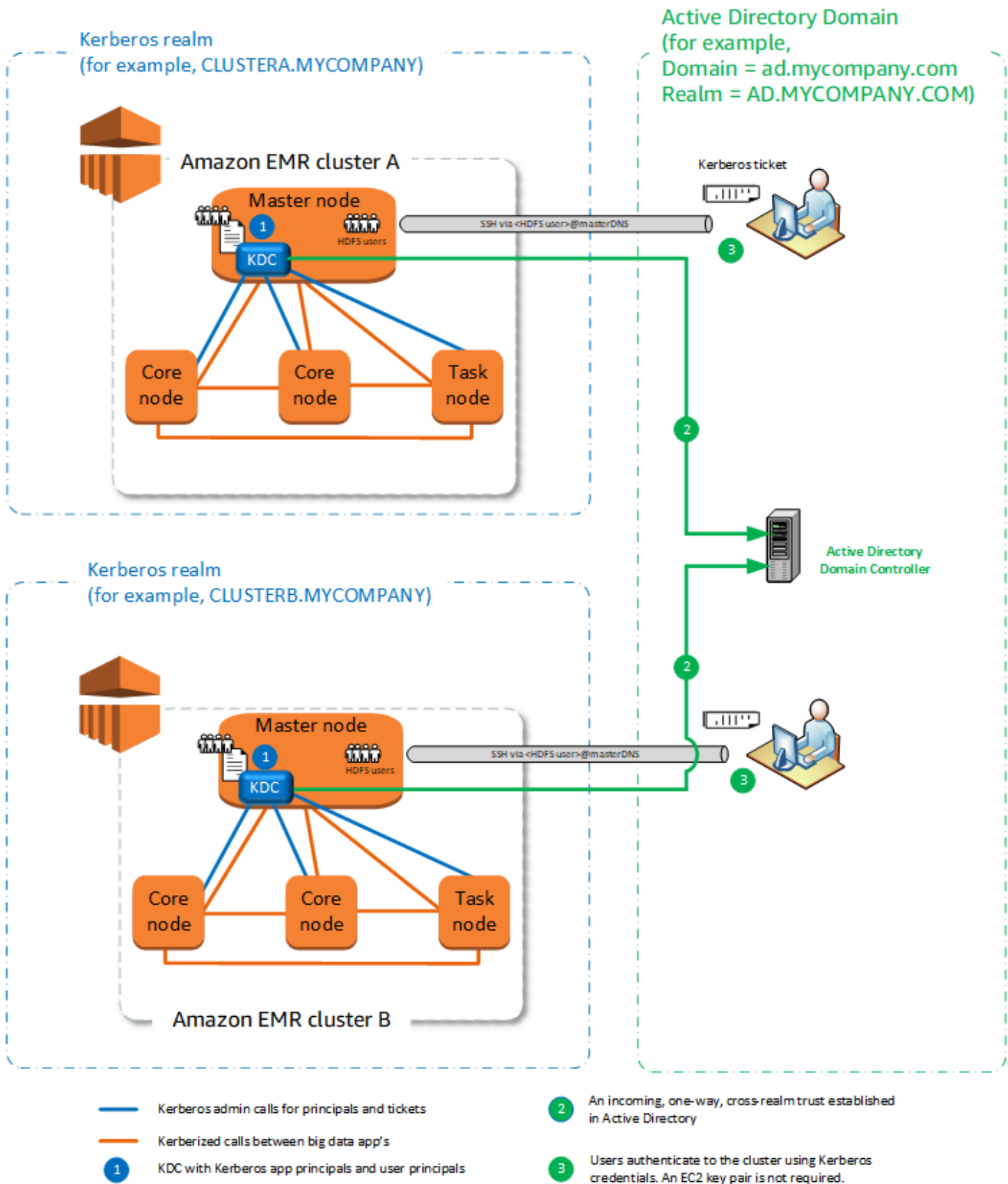
- Das KDC auf dem EMR-Cluster ist unabhängig von zentralisierten KDC-Implementierungen wie Microsoft Active Directory oder AWS Managed Microsoft AD.
- Die Auswirkungen auf die Performance sind minimal, da das KDC die Authentifizierung nur für lokale Knoten innerhalb des Clusters verwaltet.
- Optional können andere Kerberos-Cluster auf das KDC als externes KDC verweisen. Weitere Informationen finden Sie unter [Externes KDC – Primärknoten auf einem anderen Cluster](#).

Überlegungen und Einschränkungen

- Kerberos-Cluster können einander nicht authentifizieren, sodass für die Anwendungen keine Interoperabilität besteht. Wenn Cluster-Anwendungen interagieren müssen, müssen Sie eine bereichsübergreifende Vertrauensstellung zwischen Clustern oder einen Cluster als externes KDC für andere Cluster einrichten. Wenn eine realmübergreifende Vertrauensstellung eingerichtet wird, KDCs müssen sie über unterschiedliche Kerberos-Bereiche verfügen.
- Sie müssen Linux-Benutzer auf der EC2 Instanz des primären Knotens erstellen, die den KDC-Benutzerprinzipalen entsprechen, zusammen mit den HDFS-Verzeichnissen für jeden Benutzer.
- Benutzerprinzipale müssen eine EC2 private Schlüsseldatei und `kinit` Anmeldeinformationen verwenden, um über SSH eine Verbindung zum Cluster herzustellen.

Bereichsübergreifende Vertrauensstellung

In dieser Konfiguration werden Prinzipale (in der Regel Benutzer) aus einem anderen Kerberos-Bereich an den Anwendungskomponenten auf einem durch Kerberos geschützten EMR-Cluster authentifiziert, der über ein eigenes KDC verfügt. Der KDC auf dem primären Knoten baut mithilfe eines realmübergreifenden Prinzipals, der in beiden vorhanden ist, eine Vertrauensstellung mit einem anderen KDC auf. KDCs Der Name des Prinzipals und das Passwort stimmen in jedem KDC genau überein. Bereichsübergreifende Vertrauensstellungen kommen am häufigsten in Active Directory-Implementierungen vor, wie in der folgenden Abbildung dargestellt. Bereichsübergreifende Vertrauensstellungen mit einem externen MIT KDC oder einem KDC auf einem anderen Amazon-EMR Cluster werden ebenfalls unterstützt.



Vorteile

- Der EMR-Cluster, auf dem das KDC installiert ist, ist im vollständigen Besitz des KDC.
- Mit Active Directory erstellt Amazon EMR automatisch Linux-Benutzer, die Benutzerprinzipalen aus dem KDC entsprechen. Sie müssen dennoch für jeden Benutzer HDFS-Verzeichnisse erstellen. Darüber hinaus können Benutzerprinzipale in der Active Directory-Domäne mithilfe von `kinit` Anmeldeinformationen ohne die private Schlüsseldatei auf kerberisierte Cluster zugreifen. EC2 Dies beseitigt die Notwendigkeit, die Datei mit dem privaten Schlüssel für die Cluster-Benutzer freizugeben.
- Da jedes Cluster-KDC die Authentifizierung für die Knoten im Cluster verwaltet, werden die Auswirkungen der Netzwerklatenz und des Verwaltungsaufwands für eine große Anzahl an Knoten in Clustern minimiert.

Überlegungen und Einschränkungen

- Wenn Sie eine Vertrauensstellung mit einem Active-Directory-Bereich einrichten, müssen Sie beim Erstellen des Clusters Active Directory-Benutzername und -Passwort mit Berechtigungen zum Hinzufügen von Prinzipalen zur Domain angeben.
- Bereichsübergreifende Vertrauensstellungen können nicht zwischen Kerberos-Bereichen mit demselben Namen eingerichtet werden.
- Bereichsübergreifende Vertrauensstellungen müssen explizit eingerichtet werden. Beispiel: Wenn Cluster A und Cluster B eine bereichsübergreifende Vertrauensstellung mit einem KDC einrichten, vertrauen diese einander nicht inhärent und ihre Anwendungen können einander nicht authentifizieren, um miteinander zu interagieren.
- KDCs müssen unabhängig verwaltet und koordiniert werden, sodass die Anmeldeinformationen der Benutzerprinzipale exakt übereinstimmen.

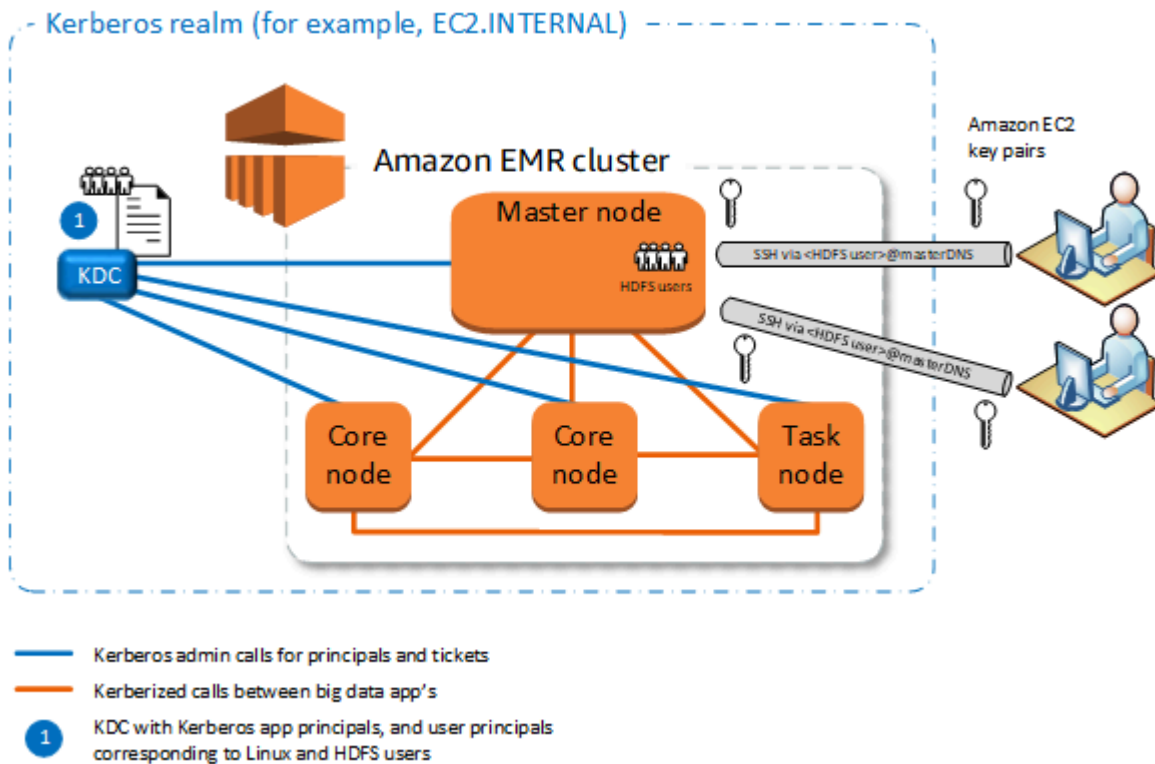
Externes KDC

Konfigurationen mit einem externen KDC werden mit Amazon EMR 5.20.0 und höher unterstützt.

- [Externes KDC – MIT KDC](#)
- [Externes KDC – Primärknoten auf einem anderen Cluster](#)
- [Externes KDC – Cluster-KDC mit anderer bereichsübergreifender Active-Directory-Vertrauensstellung](#)

Externes KDC – MIT KDC

Diese Konfiguration ermöglicht mindestens einem EMR-Cluster die Verwendung von Prinzipalen, die in einem MIT KDC-Server definiert und verwaltet werden.



Vorteile

- Die Verwaltung von Prinzipalen ist in einem einzigen KDC zusammengefasst.
- Mehrere Cluster können dasselbe KDC im selben Kerberos-Bereich verwenden. Weitere Informationen finden Sie unter [Anforderungen für die Verwendung mehrerer Cluster mit demselben KDC](#).
- Der Primärknoten auf einem durch Kerberos geschützten Cluster hat nicht die Performance-Einbußen zu verzeichnen wie der Unterhalt des KDC.

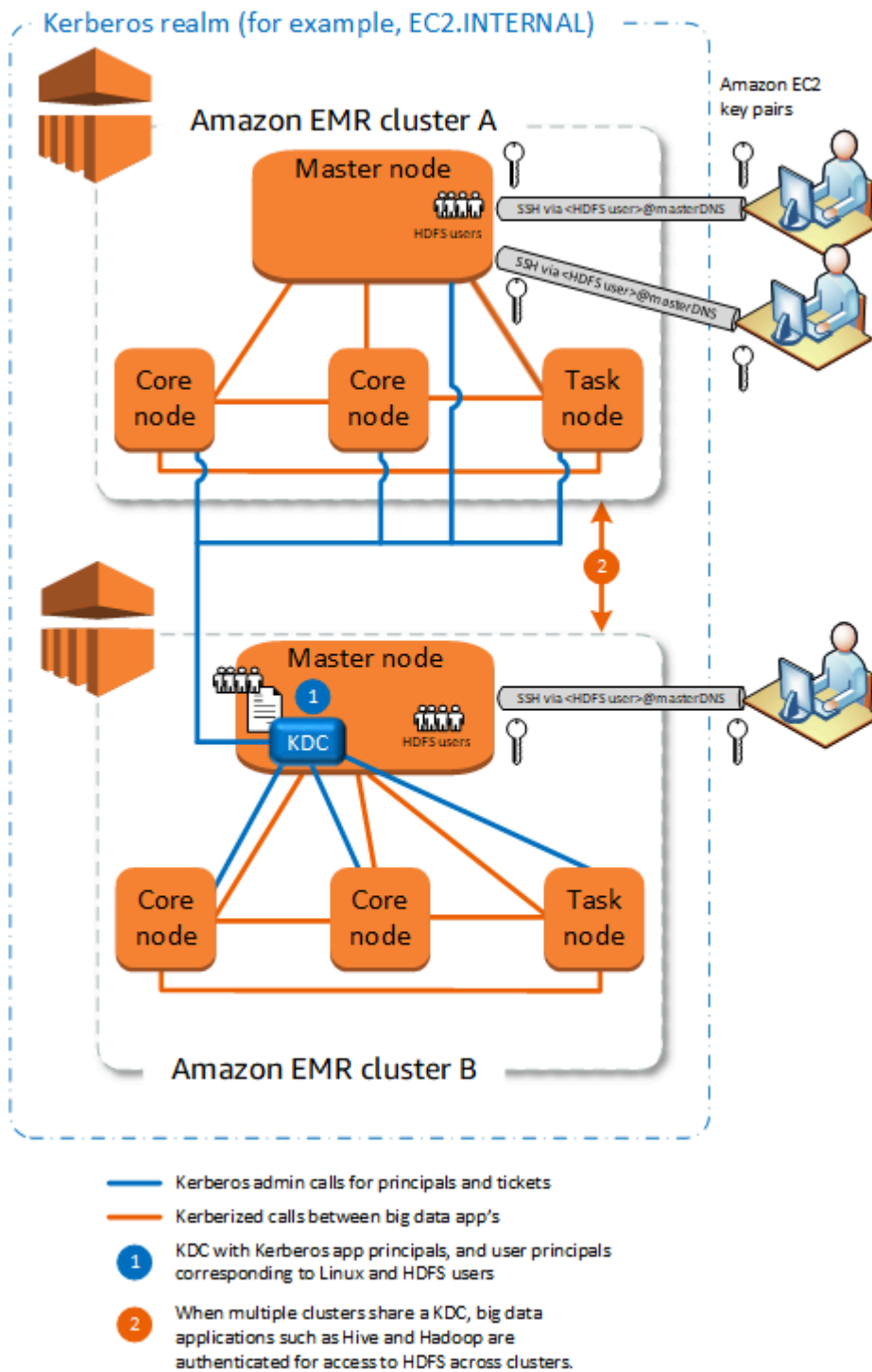
Überlegungen und Einschränkungen

- Sie müssen Linux-Benutzer auf der EC2 Instanz des primären Knotens jedes Kerberized-Clusters erstellen, die den KDC-Benutzerprinzipalen entsprechen, zusammen mit den HDFS-Verzeichnissen für jeden Benutzer.

- Benutzerprinzipale müssen eine EC2 private Schlüsseldatei und `kinit` Anmeldeinformationen verwenden, um über SSH eine Verbindung zu Kerberized-Clustern herzustellen.
- Jeder Knoten in durch Kerberos geschützten EMR-Clustern muss über eine Netzwerkroute zum KDC verfügen.
- Jeder -Knoten in durch Kerberos geschützten Clustern platziert auf dem externen KDC eine Authentifizierungshürde, sodass die Konfiguration des KDC sich auf die Cluster-Performance auswirkt. Berücksichtigen Sie bei der Konfiguration der Hardware des KDC-Servers die maximale Anzahl der Amazon-EMR-Knoten, die gleichzeitig unterstützt werden können.
- Die Cluster-Performance hängt von der Netzwerklatenz zwischen Knoten in durch Kerberos geschützten Clustern und dem KDC ab.
- Die Fehlerbehebung kann sich aufgrund von Abhängigkeiten untereinander schwieriger gestalten.

Externes KDC – Primärknoten auf einem anderen Cluster

Diese Konfiguration ist nahezu identisch mit der oben erwähnten externen MIT KDC-Implementierung, mit der Ausnahme, dass sich das KDC auf dem Primärknoten eines EMR-Clusters befindet. Weitere Informationen erhalten Sie unter [Cluster-spezifisches KDC \(KDC auf dem Primärknoten\)](#) und [Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain](#).



Vorteile

- Die Verwaltung von Prinzipalen ist in einem einzigen KDC zusammengefasst.

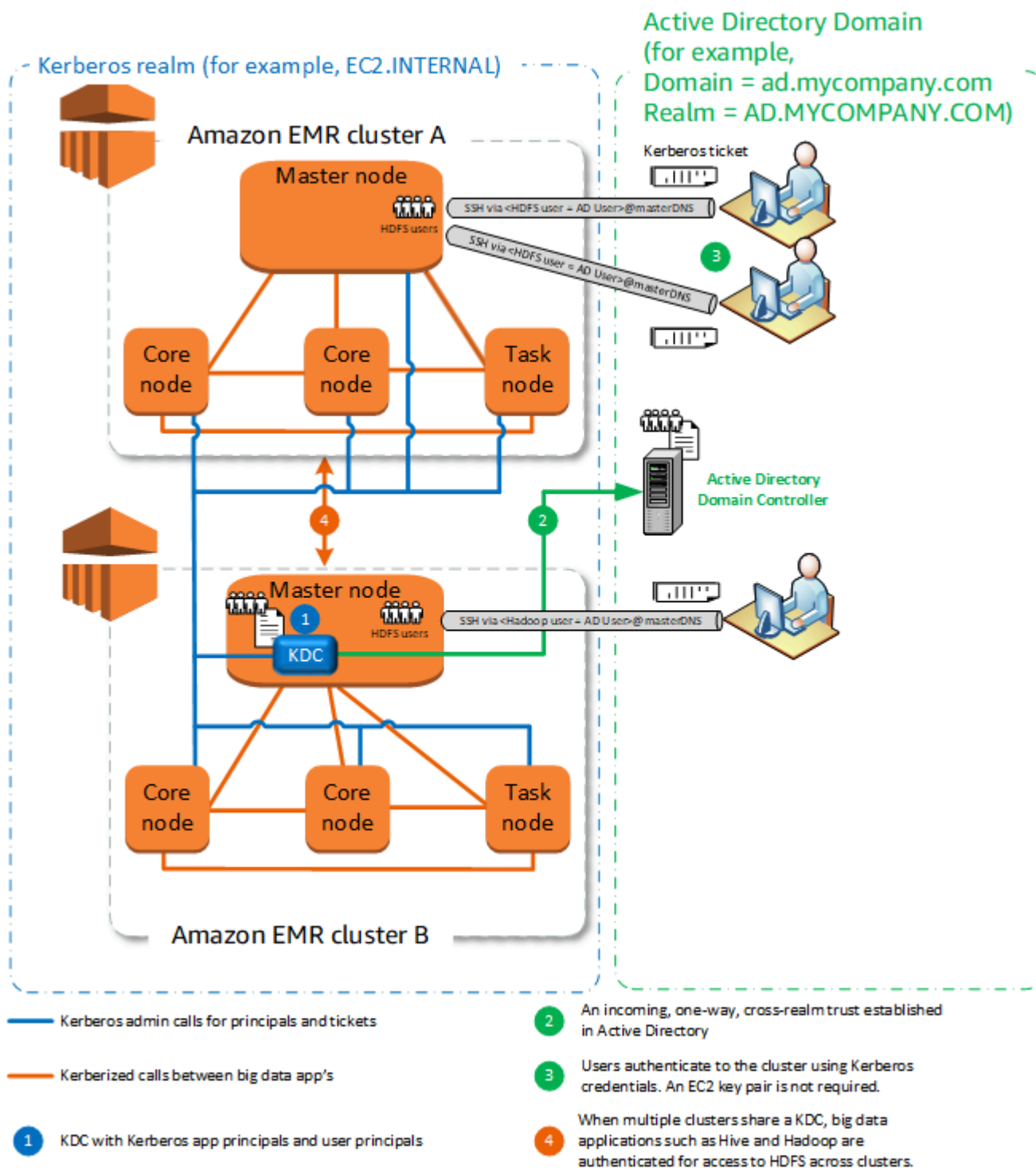
- Mehrere Cluster können dasselbe KDC im selben Kerberos-Bereich verwenden. Weitere Informationen finden Sie unter [Anforderungen für die Verwendung mehrerer Cluster mit demselben KDC](#).

Überlegungen und Einschränkungen

- Sie müssen Linux-Benutzer auf der EC2 Instanz des primären Knotens jedes Kerberized-Clusters erstellen, die den KDC-Benutzerprinzipalen entsprechen, zusammen mit den HDFS-Verzeichnissen für jeden Benutzer.
- Benutzerprinzipale müssen eine EC2 private Schlüsseldatei und `kinit` Anmeldeinformationen verwenden, um über SSH eine Verbindung zu Kerberized-Clustern herzustellen.
- Jeder Knoten in jedem EMR-Cluster muss über eine Netzwerkroute zum KDC verfügen.
- Jeder Amazon-EMR-Knoten in durch Kerberos geschützten Clustern platziert auf dem externen KDC eine Authentifizierungshürde, sodass die Konfiguration des KDC sich auf die Cluster-Performance auswirkt. Berücksichtigen Sie bei der Konfiguration der Hardware des KDC-Servers die maximale Anzahl der Amazon-EMR-Knoten, die gleichzeitig unterstützt werden können.
- Die Cluster-Performance hängt von der Netzwerklatenz zwischen Knoten in den Clustern und dem KDC ab.
- Die Fehlerbehebung kann sich aufgrund von Abhängigkeiten untereinander schwieriger gestalten.

Externes KDC – Cluster-KDC mit anderer bereichsübergreifender Active-Directory-Vertrauensstellung

Bei dieser Konfiguration erstellen Sie zunächst einen Cluster mit einem Cluster-spezifischen KDC, das eine unidirektionale bereichsübergreifende Vertrauensstellung mit Active Directory aufweist. Ein detailliertes Tutorial finden Sie unter [Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain](#). Anschließend starten Sie zusätzliche Cluster, die auf das Cluster-KDC verweisen, das die Vertrauensstellung als externes KDC besitzt. Ein Beispiel finden Sie unter [Externes Cluster-KDC mit bereichsübergreifender Active-Directory-Vertrauensstellung](#). Auf diese Weise können Sie jedem Amazon-EMR-Cluster, der das externe KDC verwendet, die Authentifizierung von Prinzipalen erlauben, die in einer Domain von Microsoft Active Directory definiert und verwaltet werden.



Vorteile

- Die Verwaltung von Prinzipalen ist in der Active-Directory-Domain zusammengefasst.

- Amazon EMR tritt dem Active-Directory-Bereich bei. Dadurch müssen keine Linux-Benutzer mehr erstellt werden, die Active-Directory-Benutzern entsprechen. Sie müssen dennoch für jeden Benutzer HDFS-Verzeichnisse erstellen.
- Mehrere Cluster können dasselbe KDC im selben Kerberos-Bereich verwenden. Weitere Informationen finden Sie unter [Anforderungen für die Verwendung mehrerer Cluster mit demselben KDC](#).
- Benutzerprinzipale in der Active Directory-Domäne können mithilfe von `kinit` Anmeldeinformationen ohne die private Schlüsseldatei auf kerberisierte Cluster zugreifen. EC2 Dies beseitigt die Notwendigkeit, die Datei mit dem privaten Schlüssel für die Cluster-Benutzer freizugeben.
- Da nur ein einziger Amazon-EMR-Primärknoten für die KDC-Verwaltung verantwortlich ist, muss nur dieser Cluster mit Active-Directory-Anmeldeinformationen erstellt werden, um die bereichsübergreifende Vertrauensstellung zwischen dem KDC und Active Directory herzustellen.

Überlegungen und Einschränkungen

- Jeder Knoten in jedem EMR-Cluster muss über eine Netzwerkroute zum KDC und den Active-Directory-Domain-Controller verfügen.
- Jeder Amazon-EMR-Knoten platziert auf dem externen KDC eine Authentifizierungshürde, sodass die Konfiguration des KDC sich auf die Cluster-Performance auswirkt. Berücksichtigen Sie bei der Konfiguration der Hardware des KDC-Servers die maximale Anzahl der Amazon-EMR-Knoten, die gleichzeitig unterstützt werden können.
- Die Cluster-Performance hängt von der Netzwerklatenz zwischen Knoten in den Clustern und dem KDC-Server ab.
- Die Fehlerbehebung kann sich aufgrund von Abhängigkeiten untereinander schwieriger gestalten.

Anforderungen für die Verwendung mehrerer Cluster mit demselben KDC

Mehrere Cluster können dasselbe KDC im selben Kerberos-Bereich verwenden. Wenn die Cluster jedoch gleichzeitig ausgeführt werden, schlagen die Cluster möglicherweise fehl, wenn sie ServicePrincipal Kerberos-Namen verwenden, die zu Konflikten führen.

Wenn Sie mehrere gleichzeitige Cluster mit demselben externen KDC haben, stellen Sie sicher, dass die Cluster unterschiedliche Kerberos-Bereiche verwenden. Wenn die Cluster denselben Kerberos-Bereich verwenden müssen, stellen Sie sicher, dass sich die Cluster in unterschiedlichen Subnetzen befinden und dass sich ihre CIDR-Bereiche nicht überschneiden.

Konfiguration von Kerberos in Amazon EMR

Dieser Abschnitt enthält Konfigurationsdetails und Beispiele für das Einrichten von Kerberos mit gängigen Architekturen. Unabhängig von der gewählten Architektur sind die Konfigurationsgrundlagen identisch und in drei Schritte unterteilt. Wenn Sie ein externes KDC verwenden oder eine bereichsübergreifende Vertrauensstellung einrichten, müssen Sie sicherstellen, dass alle Knoten in einem Cluster über eine Netzwerkroute zu einem externen KDC verfügen, einschließlich der Konfiguration der entsprechenden Sicherheitsgruppen, die den ein- und ausgehenden Kerberos-Datenverkehr erlauben.

Schritt 1: Eine Sicherheitskonfiguration mit Kerberos-Eigenschaften erstellen

Die Sicherheitskonfiguration gibt Details über das Kerberos-KDC an und ermöglicht, dass die Kerberos-Konfiguration beim Erstellen eines Clusters wiederverwendet wird. Sie können eine Sicherheitskonfiguration mithilfe der Amazon EMR-Konsole, der AWS CLI, der oder der EMR-API erstellen. Die Sicherheitskonfiguration kann auch andere Sicherheitsoptionen enthalten, wie beispielsweise die Verschlüsselung. Weitere Informationen zum Erstellen von Sicherheitskonfigurationen und Festlegen einer Sicherheitskonfiguration beim Erstellen eines Clusters finden Sie unter [Verwenden Sie Sicherheitskonfigurationen, um die Amazon EMR-Cluster-Sicherheit einzurichten](#). Informationen zu Kerberos-Eigenschaften in einer Sicherheitskonfiguration finden Sie unter [Kerberos-Einstellungen für Sicherheitskonfigurationen](#).

Schritt 2: Einen Cluster erstellen und Cluster-spezifische Kerberos-Attribute festlegen

Beim Erstellen eines Clusters legen Sie eine Kerberos-Sicherheitskonfiguration sowie Cluster-spezifische Kerberos-Optionen fest. Wenn Sie die Amazon-EMR-Konsole verwenden, sind nur die mit der angegebenen Sicherheitskonfiguration kompatiblen Kerberos-Optionen verfügbar. Wenn Sie die AWS CLI oder Amazon EMR-API verwenden, stellen Sie sicher, dass Sie Kerberos-Optionen angeben, die mit der angegebenen Sicherheitskonfiguration kompatibel sind. Beispiel: Wenn Sie beim Erstellen eines Clusters mithilfe der CLI ein Prinzipal-Passwort für eine bereichsübergreifende Vertrauensstellung verwenden und die angegebene Sicherheitskonfiguration nicht mit den Parametern der bereichsübergreifenden Vertrauensstellung konfiguriert ist, tritt ein Fehler auf. Weitere Informationen finden Sie unter [Kerberos-Einstellungen für Cluster](#).

Schritt 3: Den Cluster-Primärknoten konfigurieren

Abhängig von den Anforderungen an Ihre Architektur und Implementierung ist möglicherweise eine zusätzliche Einrichtung auf dem Cluster erforderlich. Sie können dies nach dem Erstellen oder anhand der Schritte oder Bootstrap-Aktionen während des Erstellungsvorgangs erledigen.

Für jeden Kerberos-authentifizierten Benutzer, der mittels SSH eine Verbindung mit dem Cluster herstellt, müssen Sie sicherstellen, dass Linux-Konten erstellt werden, die dem Kerberos-Benutzer entsprechen. Wenn Benutzerprinzipale von einem Active-Directory-Domain-Controller als externes KDC oder über eine bereichsübergreifende Vertrauensstellung bereitgestellt werden, erstellt Amazon EMR automatisch Linux-Benutzerkonten. Wenn Active Directory nicht verwendet wird, müssen Sie Prinzipale für jeden Benutzer erstellen, der ihrem Linux-Benutzer entspricht. Weitere Informationen finden Sie unter [Konfiguration eines Amazon EMR-Clusters für Kerberos-authentifizierte HDFS-Benutzer und SSH-Verbindungen](#).

Jeder Benutzer muss zudem über ein HDFS-Benutzerverzeichnis in seinem Besitz verfügen, das Sie erstellen müssen. Darüber hinaus muss SSH mit GSSAPI konfiguriert werden, damit Verbindungen von über Kerberos authentifizierten Benutzern zulässig sind. GSSAPI muss auf dem Primärknoten aktiviert sein und die Client-SSH-Anwendung muss so konfiguriert werden, dass sie GSSAPI verwendet. Weitere Informationen finden Sie unter [Konfiguration eines Amazon EMR-Clusters für Kerberos-authentifizierte HDFS-Benutzer und SSH-Verbindungen](#).

Sicherheitskonfiguration und Cluster-Einstellungen für Kerberos auf Amazon EMR

Wenn Sie einen durch Kerberos geschützten Cluster erstellen, geben Sie die Sicherheitskonfiguration zusammen mit den Kerberos-Attributen an, die spezifisch für den Cluster sind. Sie können eine Gruppe nicht ohne die andere angeben, sonst tritt ein Fehler auf.

Dieses Thema bietet eine Übersicht über die für Kerberos verfügbaren Konfigurationsparameter, wenn Sie eine Sicherheitskonfiguration und einen Cluster erstellen. Darüber hinaus werden CLI-Beispiele zum Erstellen von kompatiblen Sicherheitskonfigurationen und Clustern für gängige Architekturen bereitgestellt.

Kerberos-Einstellungen für Sicherheitskonfigurationen

Sie können mithilfe der Amazon EMR-Konsole, der oder der EMR-API eine Sicherheitskonfiguration erstellen AWS CLI, die Kerberos-Attribute spezifiziert. Die Sicherheitskonfiguration kann auch andere Sicherheitsoptionen enthalten, wie beispielsweise die Verschlüsselung. Weitere Informationen finden Sie unter [Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI](#).

Verwenden Sie die folgenden Referenzen, um die verfügbaren Sicherheitskonfigurationseinstellungen für die Kerberos-Architektur zu verstehen, die Sie auswählen. Die Amazon-EMR-Konsoleneinstellungen werden angezeigt. Informationen zu den entsprechenden CLI-Optionen finden Sie unter [Angaben von Kerberos-Einstellungen mit dem AWS CLI](#) oder [Beispiele für Konfigurationen](#).

Parameter		Beschreibung
Kerberos		Gibt an, dass Kerberos für Cluster aktiviert ist, die diese Sicherheitskonfiguration verwenden. Wenn ein Cluster diese Sicherheitskonfiguration verwendet, müssen für den Cluster auch Kerberos-Einstellungen angegeben sein, andernfalls tritt ein Fehler auf.
Anbieter	Cluster-dediziertes KDC	Gibt an, dass Amazon EMR einen KDC auf dem Primärknoten eines Clusters erstellt, der diese Sicherheitskonfiguration verwendet. Sie geben den Bereichsnamen und das KDC-Administratorkennwort an, wenn Sie den Cluster erstellen. Bei Bedarf können Sie von anderen Clustern aus auf diesen KDC verweisen. Erstellen Sie diese Cluster mit einer anderen Sicherheitskonfiguration, geben Sie ein externes KDC an und verwenden Sie den Bereichsnamen und das KDC-Administratorkennwort, die Sie für das clusterspezifische KDC angeben.
	Externes KDC	Nur in Amazon-EMR-Version 5.20.0 und höher verfügbar. Gibt an, dass Cluster, die diese Sicherheitskonfiguration verwenden, Kerberos-Prinzipale mithilfe eines KDC-Servers außerhalb des Clusters authentifizieren. Auf dem Cluster wird kein KDC erstellt. Sie geben den Bereichsnamen und das KDC-Administratorkennwort an, wenn Sie den Cluster erstellen.
Gültigkeitsdauer des Tickets		Optional. Gibt den Zeitraum an, für den ein vom KDC ausgestelltes Kerberos-Ticket auf Clustern gültig ist, die diese Sicherheitskonfiguration verwenden. Ticket-Gültigkeitsdauern werden aus Sicherheitsgründen beschränkt. Cluster-Anwendungen und Services verlängern Tickets automatisch, wenn sie ablaufen. Benutzer, die eine Verbindung mit dem

Parameter		Beschreibung
		Cluster über SSH mit Kerberos-Anmeldeinformationen einrichten, müssen <code>kinit</code> von der Befehlszeile des Primärknoten aus ausführen, um eine Verlängerung auszuführen, nachdem ein Ticket abgelaufen ist.
Bereichsübergreifende Vertrauensstellung		Gibt eine bereichsübergreifende Vertrauensstellung zwischen einem clusterspezifischen KDC auf Clustern, die diese Sicherheitskonfiguration verwenden, und einem KDC in einem anderen Kerberos-Bereich an. Prinzipale (in der Regel Benutzer) aus einem anderen Bereich werden gegenüber Clustern authentifiziert, die diese Konfiguration verwenden. Eine zusätzliche Konfiguration im anderen Kerberos-Bereich ist erforderlich. Weitere Informationen finden Sie unter Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain.
Realitätsübergreifende Vertrauensstellungen	Bereich	Gibt den Kerberos-Bereichsnamen des anderen Bereichs in der Vertrauensstellung an. Gemäß der Konvention sind Kerberos-Bereichsnamen mit dem Domainnamen identisch, jedoch ausschließlich in Großbuchstaben.
	Domain	Gibt den Domain-Namen des anderen Bereichs in der Vertrauensstellung an.

Parameter		Beschreibung
	Admin-Server	Gibt den Fully Qualified Domain Name (FQDN, vollständig qualifizierter Domainname) oder IP-Adresse des Admin-Servers im anderen Bereich der Vertrauensstellung an. Der Admin-Server und KDC-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, kommunizieren jedoch über andere Ports. Falls kein Port angegeben ist, wird Port 749 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :749</code>).
	KDC-Server	Gibt den vollständig qualifizierten Domain-Namen (FQDN, Fully Qualified Domain Name) oder IP-Adresse des KDC-Servers im anderen Bereich der Vertrauensstellung an. Der KDC-Server und Admin-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, nutzen jedoch andere Ports. Falls kein Port angegeben ist, wird Port 88 verwendet, da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :88</code>).
Externes KDC		Gibt an, dass das externe KDC des Clusters vom Cluster verwendet wird.

Parameter		Beschreibung	
Eigenschaften des externen KDCs	Admin-Server	Gibt den vollqualifizierten Domainnamen oder die IP-Adresse des externen Admin-Servers an. Der Admin-Server und KDC-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, kommunizieren jedoch über andere Ports. Falls kein Port angegeben ist, wird Port 749 verwendet , da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :749</code>).	
	KDC-Server	Gibt den vollqualifizierten Domainnamen des externen KDC-Servers an. Der KDC-Server und Admin-Server werden normalerweise auf demselben Rechner mit demselben FQDN ausgeführt, nutzen jedoch andere Ports. Falls kein Port angegeben ist, wird Port 88 verwendet , da es dabei um den Kerberos-Standard handelt. Sie können optional einen Port angeben (beispielsweise <code>domain.example.com :88</code>).	
	Active-Directory-Integrationen		Gibt an, dass die Kerberos-Prinzipalauthentifizierung in eine Microsoft-Active-Directory-Domain integriert ist.
	Active-Directory-Integrationseigenschaften	Active-Directory-Bereich	Gibt den Kerberos-Bereichsnamen der Active-Directory-Domain an. Gemäß der Konvention sind Kerberos-Bereichsnamen in der Regel identisch mit dem Domainnamen, jedoch ausschließlich in Großbuchstaben.
Active-Directory-Domain		Gibt den Active-Directory-Domainnamen an.	

Parameter			Beschreibung
		Active-Directory-Server	Gibt den vollqualifizierten Domainnamen des Microsoft Active Directory-Domain-Controllers an.

Kerberos-Einstellungen für Cluster

Sie können Kerberos-Einstellungen angeben, wenn Sie einen Cluster mithilfe der Amazon EMR-Konsole, der AWS CLI, der oder der EMR-API erstellen.

Verwenden Sie die folgenden Referenzen, um die verfügbaren Clusterkonfigurationseinstellungen für die Kerberos-Architektur zu verstehen, die Sie auswählen. Die Amazon-EMR-Konsoleneinstellungen werden angezeigt. Informationen zu den entsprechenden CLI-Optionen finden Sie unter [Beispiele für Konfigurationen](#).

Parameter		Beschreibung
Bereich		Der Kerberos-Bereichsname für den Cluster. Die Kerberos-Konvention ist, denselben Namen wie den Domain-Namen zu verwenden, aber in Großbuchstaben. Beispielsweise für die Domain <code>ec2.internal</code> mit <code>EC2.INTERNAL</code> als Bereichsnamen.
KDC-Administratorpasswort		Das im Cluster verwendete Passwort für <code>kadmin</code> oder <code>kadmin.local</code> . Dabei handelt es sich um Befehlszeilen-Schnittstellen zum Kerberos V5-Verwaltungssystem, das Kerberos-Prinzipale, Passwortrichtlinien und Keytabs für den Cluster verwaltet.
Prinzipal-Passwort für bereichsübergreifende Vertrauensstellungen (optional)		Erforderlich, wenn eine bereichsübergreifende Vertrauensstellung eingerichtet wird. Das Passwort für die bereichsübergreifende

Parameter	Beschreibung
	Vertrauensstellung, die über alle Bereiche hinweg identisch sein muss. Verwenden Sie ein sicheres Passwort.
Benutzer für die Verbindung mit der Active-Directory-Domain (optional)	Erforderlich bei Verwendung von Active Directory in einer bereichsübergreifenden Vertrauensstellung. Dies ist der Benutzernamendename eines Active-Directory-Kontos mit der Berechtigung, der Domain Computer hinzuzufügen. Amazon EMR verwendet diese Identität, um der Domain Cluster hinzuzufügen. Weitere Informationen finden Sie unter the section called “Schritt 3: Konten zu der Domain für den EMR-Cluster hinzufügen” .
Passwort für die Verbindung mit der Active-Directory-Domain (optional)	Das Passwort für den Benutzer für die Verbindung mit der Active-Directory-Domain. Weitere Informationen finden Sie unter the section called “Schritt 3: Konten zu der Domain für den EMR-Cluster hinzufügen” .

Beispiele für Konfigurationen

Die folgenden Beispiele veranschaulichen Sicherheitskonfigurationen und Clusterkonfigurationen für gängige Szenarien. AWS CLI Die Befehle werden der Übersichtlichkeit halber dargestellt.

Lokales KDC

Mit den folgenden Befehlen erstellen Sie einen Cluster mit einem Cluster-spezifischen KDC, das auf dem Primärknoten ausgeführt wird. Eine zusätzliche Konfiguration auf dem Cluster ist erforderlich. Weitere Informationen finden Sie unter [Konfiguration eines Amazon EMR-Clusters für Kerberos-authentifizierte HDFS-Benutzer und SSH-Verbindungen](#).

Sicherheitskonfiguration erstellen

```
aws emr create-security-configuration --name LocalKDCSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc", \
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24 }}}}'
```

Erstellen eines Clusters

```
aws emr create-cluster --release-label emr-7.9.0 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive --ec2-attributes
InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole \
--security-configuration LocalKDCSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyPassword
```

Cluster-spezifisches KDC mit bereichsübergreifender Active-Directory-Vertrauensstellung

Mit den folgenden Befehlen erstellen Sie einen Cluster mit einem Cluster-spezifischen KDC, das auf dem Primärknoten mit einer bereichsübergreifenden Vertrauensstellung an einer Active-Directory-Domain ausgeführt wird. Zusätzliche Konfiguration auf dem Cluster und in Active Directory ist erforderlich. Weitere Informationen finden Sie unter [Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain](#).

Sicherheitskonfiguration erstellen

```
aws emr create-security-configuration --name LocalKDCWithADTrustSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc", \
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24, \
"CrossRealmTrustConfiguration": {"Realm": "AD.DOMAIN.COM", \
"Domain": "ad.domain.com", "AdminServer": "ad.domain.com", \
"KdcServer": "ad.domain.com"}}}}}'
```

Erstellen eines Clusters

```
aws emr create-cluster --release-label emr-7.9.0 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration KDCWithADTrustSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyClusterKDCAdminPassword,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
```

```
CrossRealmTrustPrincipalPassword=MatchADTrustPassword
```

Externes KDC auf einem anderen Cluster

Mit den folgenden Befehlen erstellen Sie einen Cluster, der auf ein Cluster-spezifisches KDC auf dem Primärknoten eines anderen Clusters verweist, um Prinzipale zu authentifizieren. Eine zusätzliche Konfiguration auf dem Cluster ist erforderlich. Weitere Informationen finden Sie unter [Konfiguration eines Amazon EMR-Clusters für Kerberos-authentifizierte HDFS-Benutzer und SSH-Verbindungen](#).

Sicherheitskonfiguration erstellen

```
aws emr create-security-configuration --name ExtKDCOnDifferentCluster \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSOfKDCMaster:749", \
"KdcServer": "MasterDNSOfKDCMaster:88"}}}}'
```

Erstellen eines Clusters

```
aws emr create-cluster --release-label emr-7.9.0 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCOnDifferentCluster \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword
```

Externes Cluster-KDC mit bereichsübergreifender Active-Directory-Vertrauensstellung

Mit den folgenden Befehlen erstellen Sie einen Cluster ohne KDC. Der Cluster verweist auf ein Cluster-spezifisches KDC, das auf dem Primärknoten eines anderen Clusters ausgeführt wird, um Prinzipale zu authentifizieren. Dieses KDC verfügt über eine bereichsübergreifende Vertrauensstellung mit einem Active-Directory-Domain-Controller. Eine zusätzliche Konfiguration auf dem Primärknoten mit dem KDC ist erforderlich. Weitere Informationen finden Sie unter [Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain](#).

Sicherheitskonfiguration erstellen

```
aws emr create-security-configuration --name ExtKDCWithADIntegration \
--security-configuration '{"AuthenticationConfiguration": \
```

```
{
  "KerberosConfiguration": {
    "Provider": "ExternalKdc", \
    "ExternalKdcConfiguration": {
      "KdcServerType": "Single", \
      "AdminServer": "MasterDNSofClusterKDC:749", \
      "KdcServer": "MasterDNSofClusterKDC.com:88", \
      "AdIntegrationConfiguration": {
        "AdRealm": "AD.DOMAIN.COM", \
        "AdDomain": "ad.domain.com", \
        "AdServer": "ad.domain.com"
      }
    }
  }
}
```

Erstellen eines Clusters

```
aws emr create-cluster --release-label emr-7.9.0 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCWithADIntegration \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword,\
ADDomainJoinUser=MyPrivilegedADUserName,ADDomainJoinPassword=PasswordForADDomainJoinUser
```

Konfiguration eines Amazon EMR-Clusters für Kerberos-authentifizierte HDFS-Benutzer und SSH-Verbindungen

Amazon EMR erstellt Kerberos-authentifizierte Clients für Anwendungen, die auf dem Cluster ausgeführt werden, z. B. der hadoop-Benutzer, spark-Benutzer und andere. Sie können auch Benutzer hinzufügen, die mit Kerberos für Cluster-Prozesse authentifiziert werden. Authentifizierte Benutzer können dann eine Verbindung mit dem Cluster mit ihren Kerberos-Anmeldeinformationen einrichten und mit den Anwendungen arbeiten. Damit sich ein Benutzer am Cluster authentifizieren kann, sind die folgenden Konfigurationen erforderlich:

- Auf dem Cluster muss ein Linux-Konto vorhanden sein, das dem Kerberos-Prinzipal im KDC entspricht. Amazon EMR erledigt dies automatisch in Architekturen, die in Active Directory integriert sind.
- Sie müssen ein HDFS-Benutzerverzeichnis auf dem Primärknoten für jeden Benutzer erstellen und dem Benutzer Berechtigungen für das Verzeichnis erteilen.
- Sie müssen den SSH-Service konfigurieren, sodass GSSAPI auf dem Primärknoten aktiviert ist. Darüber hinaus müssen die Benutzer einen SSH-Client mit aktiviertem GSSAPI aufweisen.

Hinzufügen von Linux-Benutzern und Kerberos-Prinzipalen zum Primärknoten

Wenn Sie Active Directory nicht verwenden, müssen Sie Linux-Konten auf dem Cluster-Primärknoten erstellen und Prinzipale für diese Linux-Benutzer am KDC hinzufügen. Dies umfasst einen Prinzipal

im KDC für den Primärknoten. Zusätzlich zu den Benutzerprinzipalen erfordert das KDC, das auf dem Primärknoten ausgeführt wird, einen Prinzipal für den lokalen Host.

Wenn Ihre Architektur eine Active Directory-Integration beinhaltet, werden Linux-Benutzer und Prinzipale auf dem lokalen KDC ggf. automatisch erstellt. Sie können diesen Schritt überspringen. Weitere Informationen erhalten Sie unter [Bereichsübergreifende Vertrauensstellung](#) und [Externes KDC – Cluster-KDC mit anderer bereichsübergreifender Active-Directory-Vertrauensstellung](#).

 Important

Das KDC geht zusammen mit der Prinzipaldatenbank verloren, wenn der Primärknoten beendet wird, weil der Primärknoten kurzlebigen Speicher verwendet. Wenn Sie Benutzer für SSH-Verbindungen erstellen, empfehlen wir, eine bereichsübergreifende Vertrauensstellung mit einem externen KDC einzurichten, das für hohe Verfügbarkeit konfiguriert ist. Wenn Sie Benutzer für SSH-Verbindungen mithilfe von Linux-Konten erstellen, automatisieren Sie alternativ den Kontoerstellungsprozess mithilfe von Bootstrap-Aktionen und -Skripten, sodass er wiederholt werden kann, wenn Sie einen neuen Cluster erstellen.

Das Übermitteln eines Schritts an das Cluster nach dem Erstellen oder beim Erstellen des Clusters ist die einfachste Möglichkeit zum Hinzufügen von Benutzern und KDC-Prinzipalen. Alternativ können Sie eine Verbindung zum Primärknoten herstellen, indem Sie ein EC2 key pair als hadoop Standardbenutzer verwenden, um die Befehle auszuführen. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Im folgenden Beispiel wird einem Cluster ein bereits vorhandenes Bash-Skript `configureCluster.sh` übergeben, das auf seine Cluster-ID verweist. Das Skript wird in Amazon S3 gespeichert.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \  
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\  
Jar=s3://<region>.elasticmapreduce/libs/script-runner/script-runner.jar,\  
Args=["s3://<amzn-s3-demo-bucket>/configureCluster.sh"]
```

Das folgende Beispiel veranschaulicht den Inhalt des `configureCluster.sh`-Skripts. Das Skript wickelt zudem die Erstellung von HDFS-Benutzerverzeichnissen und die Aktivierung von GSSAPI für SSH ab, die in den folgenden Abschnitten erläutert werden.

```
#!/bin/bash
```

```
#Add a principal to the KDC for the primary node, using the primary node's returned
host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([lijuan]=pwd1 [marymajor]=pwd2 [richardroe]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
    name=${i}
    password=${arr[$i]}

    # Create a principal for each user in the primary node and require a new password
on first logon
    sudo kadmin.local -q "addprinc -pw $password +needchange $name"

    #Add hdfs directory for each user
    hdfs dfs -mkdir /user/$name

    #Change owner of each user's hdfs directory to that user
    hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

Hinzufügen von Benutzer-HDFS-Verzeichnissen

Um Ihren Benutzern zu ermöglichen, sich beim Cluster anzumelden, um Hadoop-Jobs auszuführen, müssen Sie HDFS-Benutzerverzeichnisse für ihre Linux-Konten hinzufügen und jedem Benutzer das Eigentum an ihrem Verzeichnis erteilen.

Das Übermitteln eines Schritts an das Cluster nach dem Erstellen oder beim Erstellen des Clusters ist die einfachste Möglichkeit zum Erstellen von HDFS-Verzeichnissen. Alternativ können Sie eine Verbindung zum Primärknoten herstellen, indem Sie ein EC2 key pair als hadoop Standardbenutzer verwenden, um die Befehle auszuführen. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Im folgenden Beispiel wird einem Cluster ein bereits vorhandenes Bash-Skript `AddHDFSUsers.sh` übergeben, das auf seine Cluster-ID verweist. Das Skript wird in Amazon S3 gespeichert.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://amzn-s3-demo-bucket/AddHDFSUsers.sh"]
```

Das folgende Beispiel veranschaulicht den Inhalt des `AddHDFSUsers.sh`-Skripts.

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD, or Linux users created manually on the
cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
    hdfs dfs -mkdir /user/$username
    hdfs dfs -chown $username:$username /user/$username
done
```

Aktivieren von GSSAPI für SSH

Damit für Kerberos authentifizierte Benutzer mithilfe von SSH eine Verbindung mit dem Primärknoten herstellen, muss für den SSH-Service die GSSAPI-Authentifizierung aktiviert sein. Führen Sie zum Aktivieren von GSSAPI die folgenden Befehle über die Befehlszeile des Primärknotens aus oder verwenden Sie einen Schritt zum Ausführen als Skript. Nachdem Sie SSH neu konfiguriert haben, müssen Sie den Service neu starten.

```
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```


Verwenden von SSH zum Herstellen einer Verbindung zu kerberisierten Clustern mit Amazon EMR

Dieser Abschnitt zeigt die Schritte für einen über Kerberos authentifizierten Benutzer, um eine Verbindung mit dem Primärknoten eines EMR-Clusters herzustellen.

Auf jedem Computer, der für eine SSH-Verbindung verwendet wird, müssen SSH-Client- und Kerberos-Client-Anwendungen installiert sein. Linux-Computer enthalten diese höchstwahrscheinlich standardmäßig. OpenSSH wird beispielsweise bei den meisten Linux-, Unix- und MacOS-Betriebssystemen installiert. Sie können nach einem SSH-Client suchen, indem Sie in der Befehlszeile `ssh` eingeben. Wenn Ihr Computer den Befehl nicht erkennt, installieren Sie einen SSH-Client, um eine Verbindung mit dem Primärknoten herzustellen. Das OpenSSH-Projekt bietet eine kostenlose Implementierung der umfassenden Palette von SSH-Tools. Weitere Informationen finden Sie auf der [OpenSSH](#)-Website. Windows-Benutzer können Anwendungen wie [PuTTY](#) als SSH-Client verwenden.

Weitere Informationen zu SSH-Verbindungen finden Sie unter [Eine Verbindung zu einem Amazon-EMR-Cluster herstellen](#).

SSH verwendet GSSAPI zum Authentifizieren von Kerberos-Clients und müssen die GSSAPI-Authentifizierung für den SSH-Service auf dem Cluster-Primärknoten aktivieren. Weitere Informationen finden Sie unter [Aktivieren von GSSAPI für SSH](#). SSH-Clients müssen auch GSSAPI verwenden.

MasterPublicDNS Verwenden Sie in den folgenden Beispielen den Wert, der für Master Public DNS auf der Registerkarte Zusammenfassung im Bereich Cluster-Details angezeigt wird — zum Beispiel. ***ec2-11-222-33-44.compute-1.amazonaws.com***

Voraussetzung für `krb5.conf` (nicht Active Directory)

Bei der Verwendung einer Konfiguration ohne Active-Directory-Integration muss zusätzlich zu den SSH-Client- und Kerberos-Client-Anwendungen jeder Client-Computer über eine Kopie der `/etc/krb5.conf`-Datei verfügen, die mit der `/etc/krb5.conf`-Datei auf dem Cluster-Primärknoten übereinstimmt.

So kopieren Sie die `krb5.conf`-Datei

1. Verwenden Sie SSH, um mithilfe eines EC2 key pair und des hadoop Standardbenutzers eine Verbindung zum Primärknoten herzustellen, z. B. `hadoop@MasterPublicDNS` Detaillierte Anweisungen finden Sie unter [Eine Verbindung zu einem Amazon-EMR-Cluster herstellen](#).

2. Kopieren Sie vom Primärknoten die Inhalte der `/etc/krb5.conf`-Datei. Weitere Informationen finden Sie unter [Eine Verbindung zu einem Amazon-EMR-Cluster herstellen](#).
3. Erstellen Sie auf jedem Client-Computer, der eine Verbindung zum Cluster herstellt, eine identische `/etc/krb5.conf`-Datei basierend auf der Kopie, die Sie im vorigen Schritt erstellt haben.

Verwenden von Kinit und SSH

Immer wenn ein Benutzer eine Verbindung von einem Client-Computer aus mithilfe von Kerberos-Anmeldeinformationen herstellt, muss der Benutzer zuerst Kerberos-Tickets für seinen Benutzer auf dem Client-Computer verlängern. Darüber hinaus muss der SSH-Client so konfiguriert werden, dass die GSSAPI-Authentifizierung verwendet wird.

So verwenden Sie SSH zum Herstellen einer Verbindung mit einem durch Kerberos geschützten EMR-Cluster

1. Verwenden Sie `kinit` zum Verlängern Ihres Kerberos-Tickets, wie im folgenden Beispiel gezeigt

```
kinit user1
```

2. Verwenden Sie einen `ssh`-Client zusammen mit dem Prinzipal, den Sie im Cluster-spezifischen KDC- oder Active Directory-Benutzernamen erstellt haben. Stellen Sie sicher, dass die GSSAPI-Authentifizierung aktiviert ist, wie in den folgenden Beispielen gezeigt.

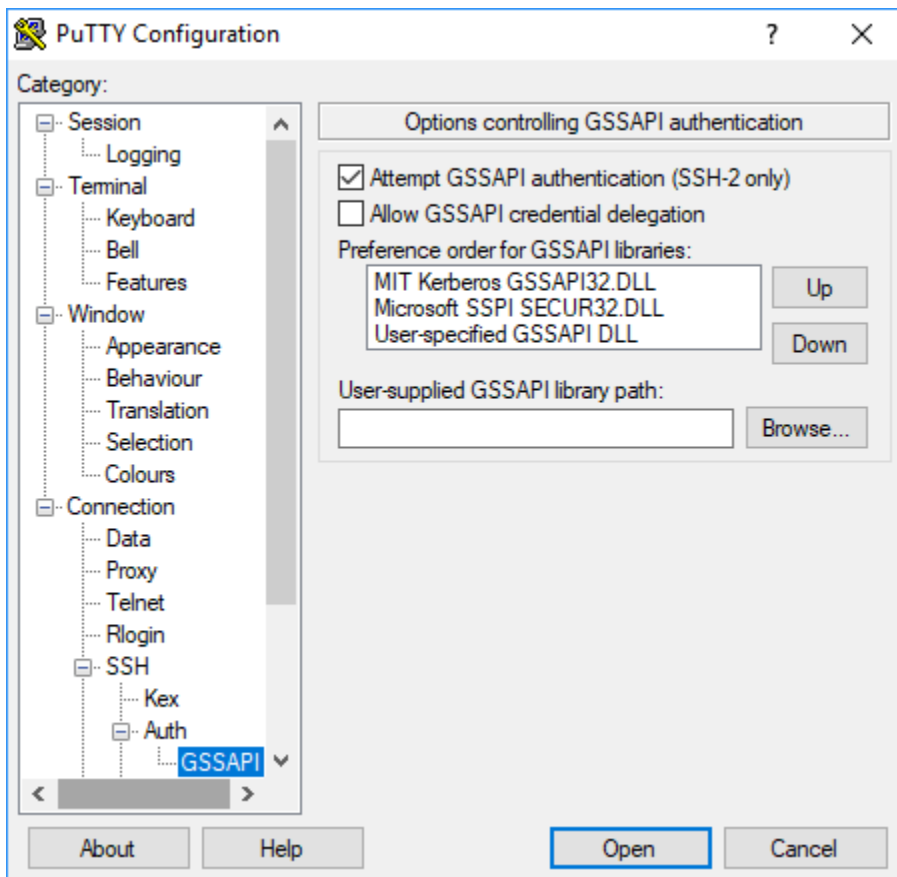
Beispiel: Linux-Benutzer

Die Option `-K` gibt die GSSAPI-Authentifizierung an.

```
ssh -K user1@MasterPublicDNS
```

Beispiel: Windows-Benutzer (PuTTY)

Stellen Sie sicher, dass die GSSAPI-Authentifizierungsoption für die Sitzung wie angezeigt aktiviert ist:



Tutorial: Konfiguration eines clusterspezifischen KDC mit Amazon EMR

Dieses Thema führt Sie durch die Erstellung eines Clusters mit einem cluster-spezifischen Key Distribution Center (KDC), das manuelle Hinzufügen von Linux-Konten zu allen Clusterknoten, das Hinzufügen von Kerberos-Prinzipalen zum KDC auf dem Primärknoten und das Sicherstellen, dass auf den Client-Computern ein Kerberos-Client installiert ist.

Weitere Informationen zur Amazon-EMR-Unterstützung für Kerberos und KDC sowie Links zur MIT-Kerberos-Dokumentation finden Sie unter [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#).

Schritt 1: Den durch Kerberos geschützten Cluster erstellen

1. Erstellen Sie eine Sicherheitskonfiguration, die Kerberos aktiviert. Das folgende Beispiel zeigt einen `create-security-configuration` Befehl mit dem AWS CLI, der die Sicherheitskonfiguration als Inline-JSON-Struktur spezifiziert. Sie können auch auf eine lokal gespeicherte Datei verweisen.

```
aws emr create-security-configuration --name MyKerberosConfig \
--security-configuration '{"AuthenticationConfiguration": {"KerberosConfiguration":
{"Provider": "ClusterDedicatedKdc", "ClusterDedicatedKdcConfiguration":
{"TicketLifetimeInHours": 24}}}}'
```

2. Erstellen Sie einen Cluster, der auf die Sicherheitskonfiguration verweist, Kerberos-Attribute für die Cluster einrichtet, und Linux-Konten unter Verwendung einer Bootstrap-Aktion hinzufügt. Das folgende Beispiel zeigt den Befehl `create-cluster` unter Verwendung der AWS CLI. Der Befehl bezieht sich auf die Sicherheitskonfiguration, die Sie oben erstellt haben, `MyKerberosConfig`. Er referenziert auch ein einfaches Skript `createlinuxusers.sh`, als Bootstrap-Aktion, das Sie erstellen und zu Amazon S3 hochladen, bevor Sie den Cluster erstellen.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-7.9.0 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair \
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd \
--bootstrap-actions Path=s3://amzn-s3-demo-bucket/createlinuxusers.sh
```

Das folgende Code zeigt den Inhalt des `createlinuxusers.sh`-Skripts, das jedem Knoten im Cluster `user1`, `user2` und `user3` hinzufügt. Im nächsten Schritt fügen Sie diese Benutzer als KDC-Prinzipale hinzu.

```
#!/bin/bash
sudo adduser user1
sudo adduser user2
sudo adduser user3
```

Schritt 2: Dem KDC Prinzipale hinzufügen, HDFS-Benutzerverzeichnisse erstellen und SSH konfigurieren

Das KDC, das auf dem Primärknoten ausgeführt wird, benötigt einen Prinzipal für den lokalen Host und für jeden Benutzer, den Sie auf dem Cluster erstellen. Sie können auch für jeden Benutzer

HDFS-Verzeichnisse erstellen, wenn sie eine Verbindung mit dem Cluster einrichten und Hadoop-Aufträge ausführen müssen. Analog dazu konfigurieren Sie den SSH-Service, um eine GSSAPI-Authentifizierung zu aktivieren, die für Kerberos erforderlich ist. Nachdem Sie GSSAPI aktiviert haben, starten Sie den SSH-Service neu.

Die einfachste Möglichkeit dafür ist, dem Cluster ein Skript zu übergeben. Das folgende Beispiel übergibt dem Cluster ein bash-Skript `configurekdc.sh`, das Sie im vorherigen Schritt erstellt haben, wobei Sie die Cluster-ID angeben. Das Skript wird in Amazon S3 gespeichert. Alternativ können Sie mithilfe eines EC2 key pair eine Verbindung zum Primärknoten herstellen, um die Befehle auszuführen oder den Schritt während der Clustererstellung abzusenden.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> --steps
  Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
  myregion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://amzn-s3-
  demo-bucket/configurekdc.sh"]
```

Das folgende Beispiel veranschaulicht den Inhalt des `configurekdc.sh`-Skripts.

```
#!/bin/bash
#Add a principal to the KDC for the primary node, using the primary node's returned
  host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=( [user1]=pwd1 [user2]=pwd2 [user3]=pwd3 )
for i in ${!arr[@]}; do
  #Assign plain language variables for clarity
  name=${i}
  password=${arr[${i}]}

  # Create principal for sshuser in the primary node and require a new password on
  first logon
  sudo kadmin.local -q "addprinc -pw $password +needchange $name"

  #Add user hdfs directory
  hdfs dfs -mkdir /user/$name

  #Change owner of user's hdfs directory to user
  hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
```

```
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/  
sshd_config  
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/  
sshd_config  
sudo systemctl restart sshd
```

Die Benutzer, die Sie hinzugefügt haben, sollten jetzt in der Lage sein, mittels SSH eine Verbindung zum Cluster herzustellen. Weitere Informationen finden Sie unter [Verwenden von SSH zum Herstellen einer Verbindung zu kerberisierten Clustern mit Amazon EMR](#).

Tutorial: Konfigurieren einer bereichsübergreifenden Vertrauensstellung mit einer Active-Directory-Domain

Wenn Sie eine bereichsübergreifende Vertrauensstellung einrichten, gestatten Sie Prinzipalen (normalerweise Benutzern) aus einem anderen Kerberos-Bereich, sich bei Anwendungskomponenten auf dem EMR-Cluster zu authentifizieren. Das dem Cluster zugeordnete Key Distribution Center (KDC) baut mithilfe eines realmübergreifenden Prinzipals, das in beiden vorhanden ist, eine Vertrauensbeziehung mit einem anderen KDC auf. KDCs Der Name des Prinzipals und das Passwort stimmen genau überein.

Eine realmübergreifende Vertrauensstellung setzt voraus, dass sie einander über das Netzwerk KDCs erreichen und die Domainnamen der jeweils anderen Seite auflösen können. Im Folgenden finden Sie Schritte zum Einrichten einer realmübergreifenden Vertrauensstellung mit einem Microsoft AD-Domänencontroller, der als EC2 Instanz ausgeführt wird, sowie ein Beispiel für eine Netzwerkkonfiguration, die die erforderliche Konnektivität und Domännennamenauflösung bereitstellt. Jede Netzwerkkonfiguration, die den erforderlichen Netzwerkverkehr zwischen KDCs den Geräten zulässt, ist zulässig.

Optional können Sie nach dem Einrichten einer bereichsübergreifenden Vertrauensstellung bei Active Directory mit einem KDC auf einem Cluster einen anderen Cluster mit einer anderen Sicherheitskonfiguration erstellen, um auf das KDC auf dem ersten Cluster als externes KDC zu verweisen. Ein Beispiel für die Einrichtung einer Sicherheitskonfiguration und eines Clusters finden Sie unter [Externes Cluster-KDC mit bereichsübergreifender Active-Directory-Vertrauensstellung](#).

Weitere Informationen zur Amazon-EMR-Unterstützung für Kerberos und KDC sowie Links zur MIT-Kerberos-Dokumentation finden Sie unter [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#).

⚠ Important

Amazon EMR unterstützt keine realmübergreifenden Vertrauensstellungen mit. AWS Directory Service for Microsoft Active Directory

[Schritt 1: Die VPC und das Subnetz einrichten](#)[Schritt 2: Den Active-Directory-Domain-Controller starten und installieren](#)[Schritt 3: Konten zu der Domain für den EMR-Cluster hinzufügen](#)[Schritt 4: Eine eingehende Vertrauensstellung auf dem Active-Directory-Domain-Controller konfigurieren](#)[Schritt 5: DHCP-Optionsmenge verwenden, um den Active-Directory-Domain-Controller zu einem VPC-DNS-Server zu machen](#)[Schritt 6: Starten eines von Kerberos geschützten EMR Clusters](#)[Schritt 7: HDFS-Benutzer erstellen und Berechtigungen für Active-Directory-Benutzerkonten in dem Cluster festlegen](#)**Schritt 1: Die VPC und das Subnetz einrichten**

Die folgenden Schritten zeigen, wie eine VPC und ein Subnetz erstellt werden, sodass das Cluster-spezifische KDC den Active-Directory-Domain-Controller erreichen und seinen Domainnamen auflösen kann. In diesen Schritten wird die Auflösung des Domain-Namens durch Angabe des Active-Directory-Domain-Controllers als Domain-Namen-Server in der DHCP-Optionsmenge durchgeführt. Weitere Informationen finden Sie unter [Schritt 5: DHCP-Optionsmenge verwenden, um den Active-Directory-Domain-Controller zu einem VPC-DNS-Server zu machen](#).

Das KDC und der Active-Directory-Domain-Controller müssen in der Lage sein, gegenseitig ihren Domainnamen aufzulösen. Dies gestattet Amazon EMR, der Domain Computer hinzuzufügen und automatisch entsprechende Linux-Konten und SSH-Parameter auf Cluster-Instances zu konfigurieren.

Wenn Amazon EMR den Domain-Namen nicht auflösen kann, können Sie unter Verwendung der IP-Adresse des Active-Directory-Domain-Controllers auf die Vertrauensstellung verweisen. Sie müssen jedoch manuell Linux-Konten hinzufügen, dem Cluster-spezifischen KDC entsprechende Prinzipale hinzufügen und SSH konfigurieren.

Einrichten der VPC und des Subnetzes

1. Erstellen Sie eine Amazon VPC mit einem einzelnen öffentlichen Subnetz. Weitere Informationen finden Sie unter [Schritt 1: Die VPC erstellen](#) im Handbuch für die ersten Schritte mit Amazon VPC.

Important

Wenn Sie einen Microsoft Active Directory-Domänencontroller verwenden, wählen Sie einen CIDR-Block für den EMR-Cluster aus, sodass alle IPv4 Adressen weniger als neun Zeichen lang sind (z. B. 10.0.0.0/16). Dies liegt daran, dass die DNS-Namen von Clustercomputern verwendet werden, wenn die Computer dem Active Directory-Verzeichnis beitreten. AWS weist [DNS-Hostnamen](#) auf der Grundlage der IPv4 Adresse zu, sodass längere IP-Adressen zu DNS-Namen mit mehr als 15 Zeichen führen können. Für Active Directory gilt ein Limit von 15 Zeichen für die Registrierung der Namen der hinzugefügten Computer, und es kürzt längere Namen, was zu unvorhersehbaren Fehlern führen kann.

2. Entfernen Sie die Standard-DHCP-Optionsmenge, die der VPC zugeordnet ist. Weitere Informationen finden Sie unter [Ändern der VPC, um keine DHCP-Optionen zu verwenden](#). Später fügen Sie eine neue hinzu, die den Active-Directory-Domain-Controller als DNS-Server spezifiziert.
3. Vergewissern Sie sich, dass DNS-Support für die VPC aktiviert ist, dass DNS-Hostnamen und DNS-Auflösung beide aktiviert sind. Standardmäßig sind sie aktiviert. Weitere Informationen finden Sie unter [Aktualisieren der DNS-Unterstützung für Ihre VPC](#).
4. Vergewissern Sie sich, dass Ihrer VPC ein Internet-Gateway zugeordnet ist. Dies ist die Standardeinstellung. Weitere Informationen finden Sie unter [Erstellen und Anfügen eines Internet-Gateways](#).

Note

In diesem Beispiel wird ein Internet-Gateway verwendet, weil Sie einen neuen Domain-Controller für den VPC einrichten. Für Ihre Anwendung ist möglicherweise kein Internet-Gateway erforderlich. Die einzige Voraussetzung ist, dass das Cluster-spezifische KDC auf den Active-Directory-Domain-Controller zugreifen kann.

5. Erstellen Sie eine benutzerdefinierte Routing-Tabelle, fügen Sie eine Route zum Internet-Gateway hinzu, und ordnen Sie ihn dann Ihrem Subnetz zu. Weitere Informationen finden Sie unter [Erstellen einer benutzerdefinierten Routing-Tabelle](#).
6. Wenn Sie die EC2 Instanz für den Domänencontroller starten, muss sie über eine statische öffentliche IPv4 Adresse verfügen, damit Sie über RDP eine Verbindung zu ihr herstellen können. Der einfachste Weg, dies zu tun, besteht darin, Ihr Subnetz so zu konfigurieren, dass öffentliche IPv4 Adressen automatisch zugewiesen werden. Dies ist nicht die Standardeinstellung, wenn ein Subnetz erstellt wird. Weitere Informationen finden Sie unter [Ändern des Attributs für die öffentliche IPv4 Adressierung Ihres Subnetzes](#). Optional können Sie die Adresse zuweisen, wenn Sie die Instance starten. Weitere Informationen finden Sie unter [Zuweisen einer öffentlichen IPv4 Adresse beim Instance-Start](#).
7. Wenn Sie fertig sind, notieren Sie sich Ihre VPC und Ihr IDs Subnetz. Sie benötigen sie später, wenn Sie den Active-Directory-Domain-Controller und den Cluster starten.

Schritt 2: Den Active-Directory-Domain-Controller starten und installieren

1. Starten Sie eine EC2 Instanz, die auf dem Microsoft Windows Server 2016 Base AMI basiert. Wir empfehlen einen m4.xlarge oder einen besseren Instance-Typ. Weitere Informationen finden Sie unter [Starten einer AWS Marketplace Instance](#) im EC2 Amazon-Benutzerhandbuch.
2. Notieren Sie sich die Gruppen-ID der Sicherheitsgruppe, die der EC2 Instance zugeordnet ist. Sie benötigen sie für [Schritt 6: Starten eines von Kerberos geschützten EMR Clusters](#). Wir verwenden `sg-012xrlmdomain345`. Alternativ können Sie verschiedene Sicherheitsgruppen für den EMR-Cluster und diese Instance angeben, die den Datenverkehr zwischen ihnen zulässt. Weitere Informationen finden Sie unter [EC2Amazon-Sicherheitsgruppen für Linux-Instances](#) im EC2 Amazon-Benutzerhandbuch.
3. Stellen Sie über RDP eine Connect zur EC2 Instanz her. Weitere Informationen finden Sie unter [Herstellen einer Verbindung zu Ihrer Windows-Instance](#) im EC2 Amazon-Benutzerhandbuch.
4. Starten Sie Server Manager, um die Domain-Services-Rolle von Active Directory auf dem Server zu installieren und zu konfigurieren. Machen Sie den Server zu einem Domain-Controller und weisen Sie einen Domain-Namen zu (wir verwenden in diesem Beispiel hier `ad.domain.com`). Notieren Sie den Domain-Namen, da Sie ihn später benötigen, wenn Sie die EMR-Sicherheitskonfiguration und den Cluster erstellen. Wenn Sie noch keine Erfahrung mit der Einrichtung von Active Directory haben, können Sie den Anweisungen in [So richten Sie Active Directory \(AD\) in Windows Server 2016 ein](#) folgen.

Die Instance startet neu, wenn Sie fertig sind.

Schritt 3: Konten zu der Domain für den EMR-Cluster hinzufügen

RDP zum Active-Directory-Domain-Controller zum Erstellen von Benutzerkonten in Active-Directory-Benutzern und -Computern für jeden Cluster-Benutzer. Weitere Informationen finden Sie unter [Erstellen eines Benutzerkontos in Active-Directory-Benutzern und -Computern](#) auf der Website Microsoft Learn. Notieren Sie den Wert für User logon name (Benutzeranmeldename) jedes Benutzers. Sie benötigen diese später, wenn Sie den Cluster konfigurieren.

Darüber hinaus erstellen Sie ein Konto mit ausreichenden Berechtigungen, um der Domain Computer hinzuzufügen. Sie geben dieses Konto an, wenn Sie einen Cluster erstellen. Amazon EMR verwendet es, um der Domain Cluster-Instances hinzuzufügen. Sie geben dieses Konto und sein Passwort in [Schritt 6: Starten eines von Kerberos geschützten EMR Clusters](#) an. Für die Delegation von Join-Berechtigungen des Computers an das Konto empfehlen wir das Erstellen einer Gruppe mit Join-Berechtigungen und die anschließende Zuweisung des Benutzers zu der Gruppe. Weitere Informationen finden Sie unter [Delegieren von Berechtigungen für den Verzeichniszugang](#) im AWS Directory Service -Administratorhandbuch.

Schritt 4: Eine eingehende Vertrauensstellung auf dem Active-Directory-Domain-Controller konfigurieren

Die folgenden Beispielbefehle erstellen ein Vertrauensverhältnis in Active Directory. Dabei handelt es sich um eine unidirektionale eingehende, nicht transitive, Bereichsvertrauensstellung mit dem Cluster-spezifischen KDC. Das Beispiel, das wir für den Bereich des Clusters verwenden, ist **EC2.INTERNAL**. Ersetzen Sie das **KDC-FQDN** durch den öffentlichen DNS-Namen, der für den Amazon EMR-Primärknoten aufgeführt ist, der den KDC hostet. Der Parameter `passwordt` gibt das cross-realm principal password (Passwort des bereichsübergreifenden Prinzipals) an, das Sie beim Erstellen eines Clusters zusammen mit dem realm (Bereich) des Clusters angeben. Der Bereichsname wird von dem Standard-Domain-Namen `us-east-1` für den Cluster abgeleitet. Die Domain ist die Active-Directory-Domain, in der Sie die Vertrauensstellung erstellen. Sie wird gemäß Konvention in Kleinbuchstaben angegeben. Im Beispiel wird **ad.domain.com** verwendet.

Öffnen Sie die Windows-Eingabeaufforderung mit Administrator-Berechtigungen und geben Sie die folgenden Befehle zum Erstellen der Vertrauensstellung auf dem Active-Directory-Domain-Controller ein:

```
C:\Users\Administrator> ksetup /addkdc EC2.INTERNAL KDC-FQDN
C:\Users\Administrator> netdom trust EC2.INTERNAL /Domain:ad.domain.com /add /realm /
passwordt:MyVeryStrongPassword
C:\Users\Administrator> ksetup /SetEncTypeAttr EC2.INTERNAL AES256-CTS-HMAC-SHA1-96
```

Schritt 5: DHCP-Optionsmenge verwenden, um den Active-Directory-Domain-Controller zu einem VPC-DNS-Server zu machen

Nachdem der Active-Directory-Domain-Controller konfiguriert ist, müssen Sie die VPC so konfigurieren, dass er als Domain-Namensserver für die Namensauflösung in Ihrer VPC verwendet wird. Dazu ordnen Sie eine DHCP-Optionsmenge zu. Geben Sie einen Wert in Domainname als Domainnamen für Ihren Cluster ein, z. B. `ec2.internal`, wenn sich Ihr Cluster in `us-east-1` befindet, oder `region.compute.internal` für andere Regionen. Für Domainnamensserver müssen Sie die IP-Adresse des Active Directory-Domänencontrollers (der vom Cluster aus erreichbar sein muss) als ersten Eintrag angeben, gefolgt von AmazonProvidedDNS (z. B. AmazonProvidedDNS). `xx.xx.xx.xx` Weitere Informationen finden Sie unter [Ändern von DHCP-Optionssätzen](#).

Schritt 6: Starten eines von Kerberos geschützten EMR Clusters

1. Erstellen Sie in Amazon EMR eine Sicherheitskonfiguration, die den Active-Directory-Domain-Controller angibt, den Sie in den vorherigen Schritten erstellt haben. Ein Beispielbefehl ist nachfolgend gezeigt. Ersetzen Sie die Domain `ad.domain.com` durch den Namen der Domain, die Sie in [Schritt 2: Den Active-Directory-Domain-Controller starten und installieren](#) angegeben haben.

```
aws emr create-security-configuration --name MyKerberosConfig \
--security-configuration '{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
          "Domain": "ad.domain.com",
          "AdminServer": "ad.domain.com",
          "KdcServer": "ad.domain.com"
        }
      }
    }
  }
}
```

2. Erstellen Sie den Cluster mit den folgenden Attributen:

- Verwenden Sie die `--security-configuration`-Option, um die Sicherheitskonfiguration anzugeben, die Sie erstellt haben. Wir verwenden `MyKerberosConfig` im Beispiel.

- Verwenden Sie die SubnetId-Eigenschaft der `--ec2-attributes` option, um das Subnetz anzugeben, das Sie in [Schritt 1: Die VPC und das Subnetz einrichten](#) erstellt haben. Wir verwenden `step1-subnet` im Beispiel.
- Verwenden Sie die `AdditionalMasterSecurityGroups` und `AdditionalSlaveSecurityGroups` der `--ec2-attributes`-Option, um anzugeben, dass die Sicherheitsgruppe, die dem AD-Domain-Controller von [Schritt 2: Den Active-Directory-Domain-Controller starten und installieren](#) zugeordnet ist, dem Cluster-Primärknoten sowie dem Core- und dem Aufgabenknoten zugeordnet ist. Wir verwenden `sg-012xrlmdomain345` im Beispiel.

Verwenden Sie `--kerberos-attributes`, um die folgenden Cluster-spezifischen Kerberos-Attribute anzugeben:

- Den Bereich für den Cluster, den Sie bei der Einrichtung des Active-Directory-Domain-Controllers angegeben haben.
- Das Prinzipal-Passwort der bereichsübergreifenden Vertrauensstellung, das Sie als `passwordt` in [Schritt 4: Eine eingehende Vertrauensstellung auf dem Active-Directory-Domain-Controller konfigurieren](#) angegeben haben.
- Ein `KdcAdminPassword`, das Sie für die Verwaltung des Cluster-spezifischen KDC verwenden können.
- Der Benutzeranmeldename und das Passwort des Active Directory-Kontos mit Computer-Join-Berechtigungen, die Sie in [Schritt 3: Konten zu der Domain für den EMR-Cluster hinzufügen](#) erstellt haben.

Das folgende Beispiel startet einen Cluster mit Schutz durch Kerberos.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-5.10.0 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair,\
SubnetId=step1-subnet, AdditionalMasterSecurityGroups=sg-012xrlmdomain345,\
AdditionalSlaveSecurityGroups=sg-012xrlmdomain345\
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
```

```
CrossRealmTrustPrincipalPassword=MatchADTrustPwd
```

Schritt 7: HDFS-Benutzer erstellen und Berechtigungen für Active-Directory-Benutzerkonten in dem Cluster festlegen

Wenn Sie eine Vertrauensstellung mit Active Directory einrichten, erstellt Amazon EMR Linux-Benutzer auf dem Cluster für jedes Active Directory-Konto. Beispielsweise hat der Benutzeranmeldename `LiJuan` in Active Directory ein Linux-Benutzerkonto von `lijuan`. Active Directory-Benutzernamen können Großbuchstaben, aber Linux ignoriert die Groß-/Kleinschreibung von Active Directory.

Um Ihren Benutzern zu ermöglichen, sich beim Cluster anzumelden, um Hadoop-Jobs auszuführen, müssen Sie HDFS-Benutzerverzeichnisse für ihre Linux-Konten hinzufügen und jedem Benutzer das Eigentum an ihrem Verzeichnis erteilen. Zu diesem Zweck empfehlen wir, dass Sie ein Skript ausführen, das Sie in Amazon S3 als Cluster-Schritt gespeichert haben. Alternativ können Sie die Befehle aus dem folgenden Skript von der Befehlszeile auf dem Primärknoten aus ausführen. Verwenden Sie das EC2 key pair, das Sie bei der Erstellung des Clusters angegeben haben, um als Hadoop-Benutzer eine Verbindung zum primären Knoten über SSH herzustellen. Weitere Informationen finden Sie unter [Verwenden Sie ein EC2 key pair für SSH-Anmeldeinformationen für Amazon EMR](#).

Führen Sie den folgenden Befehl aus, um dem Cluster, der ein Skript ausführt, einen Schritt hinzuzufügen. *AddHDFSUsers.sh*

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://amzn-s3-demo-bucket/AddHDFSUsers.sh"]
```

Der Inhalt der Datei *AddHDFSUsers.sh* lautet wie folgt.

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD or Linux users and KDC principals created
manually on the cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
```

```
# and change ownership to the user

for username in ${ADUSERS[@]}; do
    hdfs dfs -mkdir /user/$username
    hdfs dfs -chown $username:$username /user/$username
done
```

Hadoop-Gruppen zugeordnete Active-Directory-Gruppen

Amazon EMR verwendet den System Security Services Daemon (SSD), um Active Directory-Gruppen Hadoop-Gruppen zuzuordnen. Um die Gruppenzuordnungen nach der Anmeldung am Primärknoten zu bestätigen, wie in [Verwenden von SSH zum Herstellen einer Verbindung zu kerberisierten Clustern mit Amazon EMR](#) beschrieben, können Sie den Befehl `hdfs groups` ausführen, um zu bestätigen, dass Active Directory-Gruppen, zu denen Ihr Active Directory-Konto gehört, Hadoop-Gruppen für die entsprechenden Hadoop-Benutzer auf dem Cluster zugeordnet wurden. Sie können auch die Gruppenzuordnungen anderer Benutzer überprüfen, indem Sie einen oder mehrere Benutzernamen im Befehl angeben, z. B. `hdfs groups lijuan`. Weitere Informationen finden Sie unter [Groups](#) im [Apache HDFS Commands Guide](#).

Active-Directory- oder LDAP-Server für die Authentifizierung mit Amazon EMR verwenden

Mit Amazon EMR Versionen 6.12.0 und höher können Sie auch das LDAP-over-SSL-Protokoll (LDAPS) verwenden, um einen Cluster zu starten, der sich nativ in Ihren Corporate-Identity-Server integriert. LDAP (Lightweight Directory Access Protocol) ist ein offenes, herstellerneutrales Anwendungsprotokoll, das auf Daten zugreift und diese verwaltet. LDAP wird häufig für die Benutzerauthentifizierung gegen Unternehmensidentitätsserver verwendet, die auf Anwendungen wie Active Directory (AD) und OpenLDAP gehostet werden. Mit dieser nativen Integration können Sie Ihren LDAP-Server verwenden, um Benutzer auf Amazon EMR zu authentifizieren.

Zu den Highlights der Amazon EMR LDAP-Integration gehören:

- Amazon EMR konfiguriert die unterstützten Anwendungen so, dass sie sich in Ihrem Namen mit der LDAP-Authentifizierung authentifizieren.
- Amazon EMR konfiguriert und verwaltet die Sicherheit für die unterstützten Anwendungen mit dem Kerberos-Protokoll. Sie müssen keine Befehle oder Skripte eingeben.
- Sie erhalten eine detaillierte Zugriffskontrolle (FGAC) durch die Apache Ranger-Autorisierung für die Hive-Metastore-Datenbank und -Tabellen. Weitere Informationen finden Sie unter [Integrieren Sie Amazon EMR mit Apache Ranger](#).

- Wenn Sie LDAP-Anmeldeinformationen für den Zugriff auf einen Cluster benötigen, erhalten Sie eine detaillierte Zugriffskontrolle (FGAC) darüber, wer über SSH auf Ihre EMR-Cluster zugreifen kann.

Die folgenden Seiten bieten einen konzeptionellen Überblick, die Voraussetzungen und die Schritte zum Starten eines EMR-Clusters mit der Amazon-EMR-LDAP-Integration.

Themen

- [Übersicht über LDAP mit Amazon EMR](#)
- [LDAP-Komponenten für Amazon EMR](#)
- [Anwendungsunterstützung und Überlegungen zu LDAP für Amazon EMR](#)
- [Konfiguration und Start eines EMR-Clusters mit LDAP](#)
- [Beispiele für die Verwendung von LDAP mit Amazon EMR](#)

Übersicht über LDAP mit Amazon EMR

Das Lightweight Directory Access Protocol (LDAP) ist ein Softwareprotokoll, mit dem Netzwerkadministratoren den Zugriff auf Daten verwalten und kontrollieren, indem sie Benutzer im Netzwerk eines Unternehmens authentifizieren. Das LDAP-Protokoll speichert Informationen in einer hierarchischen Verzeichnisstruktur. Weitere Informationen finden Sie unter [Basiskonzepte für LDAP](#) auf LDAP.com.

Innerhalb eines Unternehmensnetzwerks verwenden viele Anwendungen möglicherweise das LDAP-Protokoll, um Benutzer zu authentifizieren. Mit der Amazon EMR LDAP-Integration können EMR-Cluster nativ dasselbe LDAP-Protokoll mit einer zusätzlichen Sicherheitskonfiguration verwenden.

Es gibt zwei Hauptimplementierungen des LDAP-Protokolls, die Amazon EMR unterstützt: Active Directory und OpenLDAP. Andere Implementierungen sind zwar möglich, aber die meisten passen zu den gleichen Authentifizierungsprotokollen wie Active Directory oder OpenLDAP.

Active Directory (AD)

Active Directory (AD) ist ein Verzeichnisservice von Microsoft für Windows-Domainnetzwerke. AD ist in den meisten Windows Server-Betriebssystemen enthalten und kann mit Clients über die Protokolle LDAP und LDAPS kommunizieren. Zur Authentifizierung versucht Amazon EMR, eine Benutzerbindung mit Ihrer AD-Instance mit dem User Principal Name (UPN – Benutzer-Prinzipal-

Name) als definiertem Namen und Passwort herzustellen. Der UPN verwendet das Standardformat `username@domain_name`.

OpenLDAP

OpenLDAP ist eine kostenlose Open-Source-Implementierung des LDAP-Protokolls. Zur Authentifizierung versucht Amazon EMR, eine Benutzerbindung mit Ihrer OpenLDAP-Instance mit dem vollständig qualifizierten Domainnamen (FQDN) als den definierten Namen und Passwort herzustellen. Der FQDN verwendet das Standardformat `username_attribute=username,LDAP_user_search_base`. In der Regel lautet der `username_attribute` Wert `uid`, und der `LDAP_user_search_base` Wert enthält die Attribute des Baums, der zum Benutzer führt. Beispiel, `ou=People,dc=example,dc=com`.

Andere kostenlose Open-Source-Implementierungen des LDAP-Protokolls folgen in der Regel einem ähnlichen FQDN wie OpenLDAP für die eindeutigen Namen ihrer Benutzer.

LDAP-Komponenten für Amazon EMR

Sie können Ihren LDAP-Server verwenden, um sich mit Amazon EMR und allen Anwendungen, die der Benutzer direkt auf dem EMR-Cluster verwendet, über die folgenden Komponenten zu authentifizieren.

Secret Agent

Der Secret Agent ist ein Cluster-Prozess, der alle Benutzeranfragen authentifiziert. Der Secret Agent erstellt die Benutzerbindung an Ihren LDAP-Server im Namen der unterstützten Anwendungen auf dem EMR-Cluster. Der Secret-Agent wird unter Benutzer `emrsecretagent` ausgeführt und schreibt Protokolle in das Verzeichnis `/emr/secretagent/log`. Diese Protokolle enthalten Details zum Status der Authentifizierungsanfrage jedes Benutzers und zu allen Fehlern, die bei der Benutzerauthentifizierung auftreten können.

System Security Services Daemon (SSSD)

SSSD ist ein Daemon, der auf jedem Knoten eines LDAP-fähigen EMR-Clusters läuft. SSSD erstellt und verwaltet einen UNIX-Benutzer, um Ihre Remote-Unternehmensidentität mit jedem Knoten zu synchronisieren. Yarn-basierte Anwendungen wie Hive und Spark erfordern, dass auf jedem Knoten, der eine Abfrage für einen Benutzer ausführt, ein lokaler UNIX-Benutzer vorhanden ist.

Anwendungsunterstützung und Überlegungen zu LDAP für Amazon EMR

In diesem Thema werden unterstützte Anwendungen, unterstützte Funktionen und nicht unterstützte Funktionen aufgeführt.

Unterstützte Anwendungen mit LDAP für Amazon EMR

Important

Die auf dieser Seite aufgeführten Anwendungen sind die einzigen Anwendungen, die Amazon EMR für LDAP unterstützt. Um die Clustersicherheit zu gewährleisten, können Sie nur LDAP-kompatible Anwendungen einbeziehen, wenn Sie einen EMR-Cluster mit aktiviertem LDAP erstellen. Wenn Sie versuchen, andere, nicht unterstützte Anwendungen zu installieren, lehnt Amazon EMR Ihre Anfrage für einen neuen Cluster ab.

Die Amazon-EMR-Versionen 6.12 und höher unterstützen die LDAP-Integration mit den folgenden Anwendungen:

- Apache Livy
- Apache Hive bis HiveServer 2 () HS2
- Trino
- Presto
- Hue

Sie können auch die folgenden Anwendungen auf einem EMR-Cluster installieren und sie so konfigurieren, dass sie Ihren Sicherheitsanforderungen entsprechen:

- Apache Spark
- Apache Hadoop

Unterstützte Features mit LDAP für Amazon EMR

Mit der LDAP-Integration können Sie die folgenden Amazon-EMR-Feature verwenden:

 Note

Um die Sicherheit der LDAP-Anmeldeinformationen zu gewährleisten, müssen Sie die Verschlüsselung während der Übertragung verwenden, um den Datenfluss innerhalb und außerhalb des Clusters zu sichern. Weitere Informationen über Verschlüsselung während der Übertragung finden Sie unter [Verschlüsseln Sie Daten im Ruhezustand und bei der Übertragung mit Amazon EMR](#).

- Verschlüsselung während der Übertragung (erforderlich) und im Ruhezustand
- Instance-Gruppen, Instance-Flotten und Spot Instances
- Neukonfiguration von Anwendungen auf einem laufenden Cluster
- Serverseitige Verschlüsselung (SSE) von EMRFS

Nicht unterstützte Funktionen

Berücksichtigen Sie die folgenden Einschränkungen, wenn Sie die Amazon- EMR-LDAP-Integration verwenden:

- Amazon EMR deaktiviert Schritte für Cluster mit aktiviertem LDAP.
- Amazon EMR unterstützt keine Runtime-Rollen und AWS Lake Formation Integrationen für Cluster mit aktiviertem LDAP.
- Amazon EMR unterstützt LDAP mit StartTLS nicht.
- Amazon EMR unterstützt keinen Hochverfügbarkeitsmodus (Cluster mit mehreren Primärknoten) für Cluster mit aktiviertem LDAP.
- Sie können Bind-Anmeldeinformationen oder Zertifikate für Cluster mit aktiviertem LDAP nicht rotieren. Wenn eines dieser Felder rotiert wurde, empfehlen wir, einen neuen Cluster mit den aktualisierten Bindungsanmeldeinformationen oder Zertifikaten zu starten.
- Bei LDAP müssen Sie exakte Suchbasen verwenden. Die LDAP-Benutzer- und Gruppensuchbasis unterstützt keine LDAP-Suchfilter.

Konfiguration und Start eines EMR-Clusters mit LDAP

In diesem Abschnitt wird beschrieben, wie Sie Amazon EMR für die Verwendung mit der LDAP-Authentifizierung konfigurieren können.

Themen

- [AWS Secrets Manager Berechtigungen zur Amazon EMR-Instance-Rolle hinzufügen](#)
- [Erstellen Sie die Amazon-EMR-Sicherheitskonfiguration für die LDAP-Integration](#)
- [Starten Sie einen EMR-Cluster, der sich mit LDAP authentifiziert](#)

AWS Secrets Manager Berechtigungen zur Amazon EMR-Instance-Rolle hinzufügen

Amazon EMR verwendet eine IAM-Servicerolle, um in Ihrem Namen Aktionen zur Bereitstellung und Verwaltung von Clustern durchzuführen. Die Servicerolle für EC2 Cluster-Instances, auch EC2 Instance-Profil für Amazon EMR genannt, ist eine spezielle Art von Servicerolle, die Amazon EMR jeder EC2 Instance in einem Cluster beim Start zuweist.

Um die Berechtigungen für einen EMR-Cluster für die Interaktion mit Amazon S3 S3-Daten und anderen AWS Diensten zu definieren, definieren Sie ein benutzerdefiniertes EC2 Amazon-Instance-Profil und nicht das, `EMR_EC2_DefaultRole` wenn Sie Ihren Cluster starten. Weitere Informationen erhalten Sie unter [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#) und [Passen Sie IAM-Rollen mit Amazon EMR an](#).

Fügen Sie dem EC2 Standard-Instance-Profil die folgenden Anweisungen hinzu, damit Amazon EMR Sitzungen taggen und auf die zugreifen kann AWS Secrets Manager , in denen LDAP-Zertifikate gespeichert sind.

```
{
  "Sid": "AllowAssumeOfRolesAndTagging",
  "Effect": "Allow",
  "Action": ["sts:TagSession", "sts:AssumeRole"],
  "Resource": [
    "arn:aws:iam::111122223333:role/LDAP_DATA_ACCESS_ROLE_NAME",
    "arn:aws:iam::111122223333:role/LDAP_USER_ACCESS_ROLE_NAME"
  ],
},
{
  "Sid": "AllowSecretsRetrieval",
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "arn:aws:secretsmanager:us-east-1:111122223333:secret:LDAP_SECRET_NAME*",
    "arn:aws:secretsmanager:us-east-1:111122223333:secret:ADMIN_LDAP_SECRET_NAME*"
  ]
}
```

```
}
```

Note

Ihre Cluster-Anfragen schlagen fehl, wenn Sie bei der Festlegung von Secrets Manager-Berechtigungen das *-Platzhalterzeichen am Ende des geheimen Namens vergessen. Der Platzhalter steht für die geheimen Versionen.

Sie sollten den Geltungsbereich der AWS Secrets Manager Richtlinie auch auf die Zertifikate beschränken, die Ihr Cluster für die Bereitstellung von Instances benötigt.

Erstellen Sie die Amazon-EMR-Sicherheitskonfiguration für die LDAP-Integration

Bevor Sie einen EMR-Cluster mit LDAP-Integration starten können, verwenden Sie die Schritte unter [Erstellen Sie eine Sicherheitskonfiguration mit der Amazon EMR-Konsole oder mit dem AWS CLI](#), um eine Amazon-EMR-Sicherheitskonfiguration für den Cluster zu erstellen. Füllen Sie die folgenden Konfigurationen im LDAPConfiguration Block unter AuthenticationConfiguration oder in den entsprechenden Feldern im Abschnitt Sicherheitskonfigurationen der Amazon-EMR-Konsole aus:

EnableLDAPAuthentication

Konsolenoption: Authentifizierungsprotokoll: LDAP

Um die LDAP-Integration zu verwenden, setzen Sie diese Option auf `true` oder wählen Sie sie als Authentifizierungsprotokoll aus, wenn Sie einen Cluster in der Konsole erstellen. Standardmäßig ist `EnableLDAPAuthentication` `true` wenn Sie eine Sicherheitskonfiguration in der Amazon-EMR-Konsole erstellen.

LDAPServerURL

Konsolenoption: Standort des LDAP-Servers

Der Speicherort des LDAP-Servers einschließlich des Präfix: `ldaps://location_of_server`.

BindCertificateARN

Konsolenoption: LDAP-SSL-Zertifikat

Der AWS Secrets Manager ARN, der das Zertifikat zum Signieren des SSL-Zertifikats enthält, das der LDAP-Server verwendet. Wenn Ihr LDAP-Server von einer öffentlichen Zertifizierungsstelle

(CA) signiert ist, können Sie einen AWS Secrets Manager ARN mit einer leeren Datei bereitstellen. Weitere Informationen zum Speichern Ihres Zertifikats in Secrets Manager finden Sie unter [Speichern Sie TLS-Zertifikate in AWS Secrets Manager](#).

BindCredentialsARN

Konsolenoption: Anmeldeinformationen: LDAP-Serverbindung

Ein AWS Secrets Manager ARN, der die Bindungsanmeldedaten für den LDAP-Administratorbenutzer enthält. Die Anmeldeinformationen werden als JSON-Objekt gespeichert. In diesem Geheimnis gibt es nur ein Schlüssel-Wert-Paar. Der Schlüssel in dem Paar ist der Benutzername und der Wert ist das Passwort. Beispiel, `{"uid=admin,cn=People,dc=example,dc=com": "AdminPassword1"}`. Dies ist ein optionales Feld, es sei denn, Sie aktivieren die SSH-Anmeldung für Ihren EMR-Cluster. In vielen Konfigurationen benötigen Active-Directory-Instances Bindungsanmeldeinformationen, damit SSSD Benutzer synchronisieren kann.

LDAPAccessFilter

Konsolenoption: LDAP-Zugriffsfilter

Gibt die Teilmenge der Objekte auf Ihrem LDAP-Server an, die sich authentifizieren können. Wenn Sie beispielsweise allen Benutzern mit der `posixAccount` Objektklasse auf Ihrem LDAP-Server Zugriff gewähren möchten, definieren Sie den Zugriffsfilter als `(objectClass=posixAccount)`.

LDAPUserSearchBase

Konsolenoption: LDAP-Benutzersuchbasis

Die Suchbasis, zu der Ihre Benutzer innerhalb Ihres LDAP-Servers gehören. Beispiel, `cn=People,dc=example,dc=com`.

LDAPGroupSearchBase

Konsolenoption: LDAP-Gruppensuchbasis

Die Suchbasis, zu der Ihre Benutzer innerhalb Ihres LDAP-Servers gehören. Beispiel, `cn=Groups,dc=example,dc=com`.

EnableSSHLogin

Konsolenoption: SSH-Anmeldung

Gibt an, ob die Kennwortauthentifizierung mit LDAP-Anmeldeinformationen zulässig ist oder nicht. Wir empfehlen nicht, dass Sie diese Option aktiviert lassen. Schlüsselpaare sind eine sicherere Route, um den Zugriff auf EMR-Cluster zu ermöglichen. Dieses Feld ist optional und standardmäßig auf `false` gesetzt.

LDAPServerType

Konsolenoption: LDAP-Servertyp

Gibt den Typ des LDAP-Servers an, mit dem Amazon EMR eine Verbindung herstellt. Unterstützte Optionen sind Active Directory und OpenLDAP. Andere LDAP-Servertypen funktionieren möglicherweise, aber Amazon EMR unterstützt offiziell keine anderen Servertypen. Weitere Informationen finden Sie unter [LDAP-Komponenten für Amazon EMR](#).

ActiveDirectoryConfigurations

Ein erforderlicher Unterblock für Sicherheitskonfigurationen, die den Active-Directory-Servertyp verwenden.

ADDomain

Konsolenoption: Active-Directory-Domain

Der Domainname, der zur Erstellung des Benutzerprinzipalnamens (UPN) für die Benutzerauthentifizierung mit Sicherheitskonfigurationen verwendet wurde, die den Active-Directory-Servertyp verwenden.

Überlegungen zu Sicherheitskonfigurationen mit LDAP und Amazon EMR

- Um eine Sicherheitskonfiguration mit Amazon-EMR-LDAP-Integration zu erstellen, müssen Sie die Verschlüsselung während der Übertragung verwenden. Informationen zur Verschlüsselung der Daten während der Übertragung finden Sie unter [Verschlüsseln Sie Daten im Ruhezustand und bei der Übertragung mit Amazon EMR](#).
- Sie können die Kerberos-Konfiguration nicht in derselben Sicherheitskonfiguration definieren. Amazon EMR stellt ein KDC bereit, das automatisch für dieses KDC reserviert ist, und verwaltet das Administratorkennwort für dieses KDC. Benutzer können nicht auf dieses Admin-Passwort zugreifen.
- Sie können keine IAM-Laufzeitrollen AWS Lake Formation in derselben Sicherheitskonfiguration definieren.
- Die `LDAPServerURL` muss das `ldaps://`-Protokoll in ihrem Wert enthalten.

- Der `LDAPAccessFilter` darf nicht leer sein.

Verwenden Sie LDAP mit der Apache-Ranger-Integration für Amazon EMR

Mit der LDAP-Integration für Amazon EMR können Sie die Integration mit Apache Ranger weiter ausbauen. Wenn Sie Ihre LDAP-Benutzer in Ranger abrufen, können Sie diese Benutzer dann einem Apache-Ranger-Richtlinien-Server zuordnen, um sie in Amazon EMR und andere Anwendungen zu integrieren. Definieren Sie dazu das `RangerConfiguration`-Feld in `AuthorizationConfiguration` der Sicherheitskonfiguration, das Sie mit Ihrem LDAP-Cluster verwenden. Weitere Informationen zum Festlegen der Sicherheitskonfiguration finden Sie unter [Erstellen einer EMR-Sicherheitskonfiguration](#).

Wenn Sie LDAP mit Amazon EMR verwenden, müssen Sie `KerberosConfiguration` mit Amazon EMR keine Integration für Apache Ranger bereitstellen.

Starten Sie einen EMR-Cluster, der sich mit LDAP authentifiziert

Führen Sie die folgenden Schritte aus, um einen EMR-Cluster mit LDAP oder Active Directory zu starten.

1. Einrichten Ihrer Umgebung:

- Stellen Sie sicher, dass die Knoten in Ihrem EMR-Cluster mit Amazon S3 kommunizieren können und AWS Secrets Manager. Weitere Informationen darüber, wie Sie Ihre EC2 Instance-Profilrolle ändern können, um mit diesen Diensten zu kommunizieren, finden Sie unter [AWS Secrets Manager Berechtigungen zur Amazon EMR-Instance-Rolle hinzufügen](#).
- Wenn Sie Ihren EMR-Cluster in einem privaten Subnetz ausführen möchten, sollten Sie Amazon VPC-Endpunkte oder Network Address Translation (NAT) verwenden, um die VPC für die Kommunikation mit S3 AWS PrivateLink und Secrets Manager zu konfigurieren. Weitere Informationen finden Sie unter [AWS PrivateLink und VPC-Endpoints](#) und [NAT-Instances](#) im Handbuch für die ersten Schritte mit Amazon VPC.
- Stellen Sie sicher, dass zwischen Ihrem EMR-Cluster und dem LDAP-Server eine Netzwerkverbindung besteht. Ihre EMR-Cluster müssen über das Netzwerk auf Ihren LDAP-Server zugreifen. Die Primär-, Kern- und Aufgabenknoten für den Cluster kommunizieren mit dem LDAP-Server, um Benutzerdaten zu synchronisieren. Wenn Ihr LDAP-Server auf Amazon läuft EC2, aktualisieren Sie die EC2 Sicherheitsgruppe so, dass sie Datenverkehr vom EMR-Cluster akzeptiert. Weitere Informationen finden Sie unter [AWS Secrets Manager Berechtigungen zur Amazon EMR-Instance-Rolle hinzufügen](#).

2. Erstellen Sie eine Amazon-EMR-Sicherheitskonfiguration für die LDAP-Integration. Weitere Informationen finden Sie unter [Erstellen Sie die Amazon-EMR-Sicherheitskonfiguration für die LDAP-Integration](#).
3. Nachdem Sie nun eingerichtet sind, gehen Sie wie unter [Starten eines Amazon-EMR-Clusters](#) beschrieben vor, um Ihren Cluster mit den folgenden Konfigurationen zu starten:
 - Wählen Sie Amazon EMR Version 6.12. oder höher aus. Es wird empfohlen, ein Upgrade auf die neueste Amazon-EMR-Version durchzuführen.
 - Geben Sie für Ihren Cluster nur Anwendungen an oder wählen Sie sie aus, die LDAP unterstützen. Eine Liste der LDAP-unterstützten Anwendungen mit Amazon EMR finden Sie unter [Anwendungsunterstützung und Überlegungen zu LDAP für Amazon EMR](#).
 - Verwenden Sie die Sicherheitskonfiguration, die Sie im vorherigen Schritt erstellt haben.

Beispiele für die Verwendung von LDAP mit Amazon EMR

Sobald Sie [einen EMR-Cluster bereitgestellt haben, der die LDAP-Integration verwendet](#), können Sie Ihre LDAP-Anmeldeinformationen über den integrierten Authentifizierungsmechanismus für Benutzernamen und Passwörter für jede [unterstützte Anwendung](#) bereitstellen. Diese Seite zeigt einige Beispiele.

Verwendung der LDAP-Authentifizierung mit Apache Hive

Example – Apache Hive

Der folgende Beispielbefehl startet eine Apache Hive-Sitzung über HiveServer 2 und Beeline:

```
beeline -u "jdbc:hive2://$HOSTNAME:10000/default;ssl=true;sslTrustStore=
$TRUSTSTORE_PATH;trustStorePassword=$TRUSTSTORE_PASS" -n LDAP_USERNAME -
p LDAP_PASSWORD
```

Verwendung der LDAP-Authentifizierung mit Apache Hive

Example – Apache Livy

Der folgende Beispielbefehl startet eine Livy-Sitzung über cURL. Ersetzen Sie **ENCODED-KEYPAIR** durch eine Base64-kodierte Zeichenfolge für `username:password`.


```
curl -X POST --data '{"proxyUser":"LDAP_USERNAME","kind": "pyspark"}' -H "Content-Type: application/json" -H "Authorization: Basic ENCODED-KEYPAIR" DNS_OF_PRIMARY_NODE:8998/sessions
```

Verwenden der LDAP-Authentifizierung mit Presto

Example – Presto

Der folgende Beispielbefehl startet eine Presto-Sitzung über die Presto-CLI:

```
presto-cli --user "LDAP_USERNAME" --password --catalog hive
```

Nachdem Sie diesen Befehl ausgeführt haben, geben Sie das LDAP-Passwort an der Eingabeaufforderung ein.

Verwenden der LDAP-Authentifizierung mit Trino

Example – Trino

Der folgende Beispielbefehl startet eine Presto-Sitzung über die Presto-CLI:

```
trino-cli --user "LDAP_USERNAME" --password --catalog hive
```

Nachdem Sie diesen Befehl ausgeführt haben, geben Sie das LDAP-Passwort an der Eingabeaufforderung ein.

Verwenden der LDAP-Authentifizierung mit Hue

Sie können über einen SSH-Tunnel, den Sie auf dem Cluster erstellen, auf die Hue-Benutzeroberfläche zugreifen, oder Sie können einen Proxy-Server einrichten, der die Verbindung öffentlich an Hue überträgt. Da Hue standardmäßig nicht im HTTPS-Modus läuft, empfehlen wir die Verwendung einer zusätzlichen Verschlüsselungsebene, um sicherzustellen, dass die Kommunikation zwischen den Clients und der Hue-Benutzeroberfläche mit HTTPS verschlüsselt ist. Dadurch wird die Wahrscheinlichkeit verringert, dass Sie versehentlich Benutzeranmeldeinformationen im Klartext preisgeben.

Um die Hue-Benutzeroberfläche zu verwenden, öffnen Sie die Hue-Benutzeroberfläche in Ihrem Browser und geben Sie Ihren LDAP-Benutzernamen und das Passwort ein, um sich anzumelden.

Wenn die Anmeldeinformationen korrekt sind, meldet Hue Sie an und verwendet Ihre Identität, um Sie bei allen unterstützten Anwendungen zu authentifizieren.

Verwendung von SSH für die Passwortauthentifizierung und Kerberos-Tickets für andere Anwendungen

 Important

Wir empfehlen nicht, die Passwortauthentifizierung für SSH in einen EMR-Cluster zu verwenden.

Sie können Ihre LDAP-Anmeldeinformationen verwenden, um eine SSH-Verbindung zu einem EMR-Cluster herzustellen. Stellen Sie dazu die `EnableSSHLogin`-Konfiguration in der Amazon-EMR-Sicherheitskonfiguration, die Sie zum Starten des Clusters verwenden, auf `true` ein. Verwenden Sie dann den folgenden Befehl, um per SSH auf den Cluster zuzugreifen, sobald dieser gestartet wurde:

```
ssh username@EMR_PRIMARY_DNS_NAME
```

Nachdem Sie diesen Befehl ausgeführt haben, geben Sie das LDAP-Passwort an der Eingabeaufforderung ein.

Amazon EMR enthält ein Cluster-Skript, das es Benutzern ermöglicht, eine Kerberos-Keytab-Datei und ein Ticket für die Verwendung mit unterstützten Anwendungen zu generieren, die LDAP-Anmeldeinformationen nicht direkt akzeptieren. Einige dieser Anwendungen umfassen `spark-submit` Spark SQL und. PySpark

Führen Sie `ldap-kinit` aus und befolgen Sie die Eingabeaufforderungen. Wenn die Authentifizierung erfolgreich ist, wird die Kerberos-Keytab-Datei mit einem gültigen Kerberos-Ticket in Ihrem Home-Verzeichnis angezeigt. Verwenden Sie das Kerberos-Ticket, um Anwendungen wie in jeder Kerberized-Umgebung auszuführen.

Integrieren Sie Amazon EMR mit AWS IAM Identity Center

Mit Amazon EMR-Versionen 6.15.0 und höher können Sie Identitäten von verwenden, um sich bei einem Amazon AWS IAM Identity Center EMR-Cluster zu authentifizieren. Die folgenden Abschnitte bieten einen konzeptionellen Überblick, die Voraussetzungen und die Schritte zum Starten eines EMR-Clusters mit der Identity-Center-Integration.

Themen

- [Übersicht](#)
- [Features und Vorteile](#)
- [Erste Schritte mit der AWS IAM Identity Center Integration für Amazon EMR](#)
- [Überlegungen und Einschränkungen für Amazon EMR mit Identity-Center-Integration](#)

Übersicht

[Identity Center](#) ist der empfohlene Ansatz für die Authentifizierung und Autorisierung von Mitarbeitern AWS für Unternehmen jeder Größe und Art. Mit Identity Center können Sie Benutzeridentitäten in Ihrer vorhandenen Identitätsquelle AWS, einschließlich Microsoft Active Directory, Okta, Ping Identity, Google Workspace und Microsoft Entra ID (ehemals Azure AD) JumpCloud, erstellen und verwalten oder eine Verbindung zu Ihrer vorhandenen Identitätsquelle herstellen.

[Die Weitergabe vertrauenswürdiger Identitäten](#) ist eine AWS IAM Identity Center Funktion, mit der Administratoren von Connected den Zugriff auf Servicedaten gewähren und prüfen AWS-Services können. Der Zugriff auf diese Daten basiert auf Benutzerattributen wie Gruppenzuordnungen. Die Einrichtung von Trusted Identity Propagation erfordert die Zusammenarbeit zwischen den Administratoren von Connected AWS-Services und den IAM Identity Center-Administratoren. Weitere Informationen finden Sie unter [Voraussetzungen und Überlegungen](#).

Features und Vorteile

Die Integration von Amazon EMR in IAM Identity Center bietet die folgenden Vorteile:

- Amazon EMR stellt Anmeldeinformationen bereit, um Ihre Identity-Center-Identität an einen EMR-Cluster weiterzuleiten.
- Amazon EMR konfiguriert alle unterstützten Anwendungen für die Authentifizierung mit den Cluster-Anmeldeinformationen.
- Amazon EMR konfiguriert und verwaltet die Sicherheit für die unterstützten Anwendungen mit dem Kerberos-Protokoll und ohne dass Befehle oder Skripts von Ihnen erforderlich sind.
- Die Möglichkeit, die Amazon-S3-Autorisierung auf Präfixebene mit Identity-Center-Identitäten auf von S3 Access Grants verwalteten S3-Präfixen durchzusetzen.
- Die Möglichkeit, die Autorisierung auf Tabellenebene mit Identity Center-Identitäten für AWS Lake Formation verwaltete AWS Glue-Tabellen durchzusetzen.

Erste Schritte mit der AWS IAM Identity Center Integration für Amazon EMR

Dieser Abschnitt hilft Ihnen bei der Konfiguration von Amazon EMR für die Integration mit AWS IAM Identity Center.

Themen

- [Eine Identity-Center-Instance erstellen](#)
- [Eine IAM-Rolle für Identity Center erstellen](#)
- [Fügen Sie Berechtigungen für Dienste hinzu, die nicht in IAM Identity Center integriert sind](#)
- [Eine Identity-Center-fähige Sicherheitskonfiguration erstellen](#)
- [Einen Identity-Center-fähigen Cluster erstellen und starten](#)
- [Lake Formation für einen IAM-Identity-Center-aktivierten EMR-Cluster konfigurieren](#)
- [Mit S3 Access Grants auf einem IAM-Identity-Center-fähigen EMR-Cluster arbeiten](#)

Note

Um die Identity Center-Integration mit EMR, Lake Formation oder S3 nutzen zu können, müssen Access Grants aktiviert sein. Sie können auch beide verwenden. Wenn keines der beiden aktiviert ist, wird die Identity Center-Integration nicht unterstützt.

Eine Identity-Center-Instance erstellen

Wenn Sie noch keine haben, erstellen Sie eine Identity-Center-Instance in der AWS-Region , in der Sie Ihren EMR-Cluster starten möchten. Eine Identity-Center-Instance kann nur in einer einzigen Region für ein AWS-Konto existieren.

Verwenden Sie den folgenden AWS CLI Befehl, um eine neue Instanz mit dem Namen zu erstellen *MyInstance*:

```
aws sso-admin create-instance --name MyInstance
```

Eine IAM-Rolle für Identity Center erstellen

Um Amazon EMR zu integrieren AWS IAM Identity Center, erstellen Sie eine IAM-Rolle, die sich über den EMR-Cluster bei Identity Center authentifiziert. Amazon EMR verwendet im Hintergrund SigV4-Anmeldeinformationen, um die Identity-Center-Identität an nachgelagerte Services weiterzuleiten,

wie z. B. AWS Lake Formation. Ihre Rolle sollte auch über die entsprechenden Berechtigungen zum Aufrufen der nachgelagerten Services verfügen.

Verwenden Sie die folgende Berechtigungsrichtlinie zum Erstellen der Rolle:

```
{
  "Statement": [
    {
      "Sid": "IdCPermissions",
      "Effect": "Allow",
      "Action": [
        "sso-oauth:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "GlueandLakePermissions",
      "Effect": "Allow",
      "Action": [
        "glue:*",
        "lakeformation:GetDataAccess"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AccessGrantsPermissions",
      "Effect": "Allow",
      "Action": [
        "s3:GetDataAccess",
        "s3:GetAccessGrantsInstanceForPrefix"
      ],
      "Resource": "*"
    }
  ]
}
```

Die Vertrauensrichtlinie für diese Rolle ermöglicht es der InstanceProfile-Rolle, sie die Rolle übernehmen zu lassen.

```
{
  "Sid": "AssumeRole",
  "Effect": "Allow",
  "Principal": {
```

```

    "AWS": "arn:aws:iam::12345678912:role/EMR_EC2_DefaultRole"
  },
  "Action": [
    "sts:AssumeRole",
    "sts:SetContext"
  ]
}

```

Wenn die Rolle keine vertrauenswürdigen Anmeldeinformationen hat und auf eine durch Lake Formation geschützte Tabelle zugreift, setzt Amazon EMR den Wert `principalId` der angenommenen Rolle automatisch auf `userID-untrusted`. Im Folgenden finden Sie einen Auszug aus einem Ereignis, das den anzeigt. CloudTrail `principalId`

```

{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ABCDEFGH1JKLMN02PQR3TU:5000-untrusted",
    "arn": "arn:aws:sts::123456789012:assumed-role/EMR_TIP/5000-untrusted",
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGH1IJKLMNOPQ7R3"
    ...
  }
}

```

Fügen Sie Berechtigungen für Dienste hinzu, die nicht in IAM Identity Center integriert sind

AWS Anmeldeinformationen, die Trusted Identity Propagation die in der IAM-Rolle definierten IAM-Richtlinien für alle Aufrufe von Diensten verwenden, die nicht in IAM Identity Center integriert sind. Dazu gehören beispielsweise die AWS Key Management Service Ihre Rolle sollte auch alle IAM-Berechtigungen für solche Dienste definieren, auf die Sie zugreifen möchten. Zu den derzeit unterstützten integrierten Diensten von IAM Identity Center gehören AWS Lake Formation Amazon S3 Access Grants.

Weitere Informationen zu Trusted Identity Propagation finden Sie unter [Trusted Identity Propagation zwischen Anwendungen](#).

Eine Identity-Center-fähige Sicherheitskonfiguration erstellen

Um einen EMR-Cluster mit IAM-Identity-Center-Integration zu starten, verwenden Sie den folgenden Beispielbefehl, um eine Amazon-EMR-Sicherheitskonfiguration zu erstellen, für die Identity Center aktiviert ist. Jede Konfiguration wird im Folgenden erklärt.

```
aws emr create-security-configuration --name "IdentityCenterConfiguration-with-lf-
accessgrants" --region "us-west-2" --security-configuration '{
  "AuthenticationConfiguration":{
    "IdentityCenterConfiguration":{
      "EnableIdentityCenter":true,
      "IdentityCenterApplicationAssignmentRequired":false,
      "IdentityCenterInstanceARN": "arn:aws:sso::instance/
ssoins-123xxxxxxxxxx789"
    }
  },
  "AuthorizationConfiguration": {
    "LakeFormationConfiguration": {
      "AuthorizedSessionTagValue": "Amazon EMR"
    },
    "IAMConfiguration": {
      "EnableApplicationScopedIAMRole": true,
      "ApplicationScopedIAMRoleConfiguration": {
        "PropagateSourceIdentity": true
      }
    }
  },
  "EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "EnableAtRestEncryption": false,
    "InTransitEncryptionConfiguration": {
      "TLSCertificateConfiguration": {
        "CertificateProviderType": "PEM",
        "S3Object": "s3://amzn-s3-demo-bucket/cert/my-certs.zip"
      }
    }
  }
}'
```

- **EnableIdentityCenter** – (erforderlich) Aktiviert die Identity-Center-Integration.
- **IdentityCenterInstanceARN**— (optional) Der ARN der Identity Center-Instanz. Wenn dies nicht enthalten ist, wird der bestehende ARN der IAM Identity Center-Instanz als Teil des Konfigurationsschritts gesucht.
- **IAMRoleForEMRIdentityCenterApplicationARN** – (erforderlich) Die IAM-Rolle, die Identity-Center-Token aus dem Cluster bezieht.

- **IdentityCenterApplicationAssignmentRequired** – (boolean) Legt fest, ob für die Nutzung der Identity-Center-Anwendung eine Zuweisung erforderlich ist. Dies ist ein optionales Feld. Wenn kein Wert angegeben wird, ist `false` der Standardwert.
- **AuthorizationConfiguration/LakeFormationConfiguration**— Konfigurieren Sie optional die Autorisierung:
 - **IAMConfiguration**— Ermöglicht die Verwendung der Funktion EMR Runtime Roles zusätzlich zu Ihrer TIP-Identität. Wenn Sie diese Konfiguration aktivieren, müssen Sie (oder der AWS Anrufer-Service) bei jedem Aufruf der EMR Steps oder EMR eine IAM-Laufzeitrolle angeben. `GetClusterSessionCredentials` APIs Wenn der EMR-Cluster mit SageMaker Unified Studio verwendet wird, ist diese Option erforderlich, wenn Trusted Identity Propagation ebenfalls aktiviert ist.
 - **EnableLakeFormation** – Aktivieren Sie die Lake-Formation-Autorisierung auf dem Cluster.

Um die Identity-Center-Integration mit Amazon EMR zu aktivieren, müssen Sie `EncryptionConfiguration` und `IntransitEncryptionConfiguration` angeben.

Einen Identity-Center-fähigen Cluster erstellen und starten

Nachdem Sie nun die IAM-Rolle eingerichtet haben, die sich bei Identity Center authentifiziert, und eine Amazon-EMR-Sicherheitskonfiguration erstellt haben, für die Identity Center aktiviert ist, können Sie Ihren identitätsbewussten Cluster erstellen und starten. Schritte zum Starten Ihres Clusters mit der erforderlichen Sicherheitskonfiguration finden Sie unter [Geben Sie eine Sicherheitskonfiguration für einen Amazon EMR-Cluster an](#).

In den folgenden Abschnitten wird beschrieben, wie Sie Ihren Identity Center-fähigen Cluster mit Sicherheitsoptionen konfigurieren, die Amazon EMR unterstützt:

- [Mit S3 Access Grants auf einem IAM-Identity-Center-fähigen EMR-Cluster arbeiten](#)
- [Lake Formation für einen IAM-Identity-Center-aktivierten EMR-Cluster konfigurieren](#)

Lake Formation für einen IAM-Identity-Center-aktivierten EMR-Cluster konfigurieren

Sie können die Integration in [AWS Lake Formation](#) Ihren AWS IAM Identity Center aktivierten EMR-Cluster durchführen.

Stellen Sie zunächst sicher, dass Sie eine Identity-Center-Instance in derselben Region wie Ihr Cluster eingerichtet haben. Weitere Informationen finden Sie unter [Eine Identity-Center-Instance](#)

[erstellen](#). Sie finden den Instance-ARN in der IAM-Identity-Center-Konsole, wenn Sie die Instance-Details anzeigen, oder über den folgenden Befehl, mit dem Details für all Ihre Instances über die CLI angezeigt werden:

```
aws sso-admin list-instances
```

Verwenden Sie dann den ARN und Ihre AWS Konto-ID mit dem folgenden Befehl, um Lake Formation so zu konfigurieren, dass es mit IAM Identity Center kompatibel ist:

```
aws lakeformation create-lake-formation-identity-center-configuration --cli-input-json
file://create-lake-fromation-idc-config.json
json input:
{
  "CatalogId": "account-id/org-account-id",
  "InstanceArn": "identity-center-instance-arn"
}
```

Rufen Sie jetzt `put-data-lake-settings` auf und aktivieren Sie `AllowFullTableExternalDataAccess` mit Lake Formation:

```
aws lakeformation put-data-lake-settings --cli-input-json file://put-data-lake-
settings.json
json input:
{
  "DataLakeSettings": {
    "DataLakeAdmins": [
      {
        "DataLakePrincipalIdentifier": "admin-ARN"
      }
    ],
    "CreateDatabaseDefaultPermissions": [...],
    "CreateTableDefaultPermissions": [...],
    "AllowExternalDataFiltering": true,
    "AllowFullTableExternalDataAccess": true
  }
}
```

Erteilen Sie abschließend dem Identity-ARN für den Benutzer, der auf den EMR-Cluster zugreift, vollständige Tabellenberechtigungen. Der ARN enthält die Benutzer-ID von Identity Center. Navigieren Sie in der Konsole zu Identity Center, wählen Sie Users (Benutzer) und dann den Benutzer aus, um dessen allgemeine Informationseinstellungen anzuzeigen.

Kopieren Sie die Benutzer-ID und fügen Sie sie für *user-id* in den folgenden ARN ein:

```
arn:aws:identitystore:::user/user-id
```

 Note

Abfragen auf dem EMR-Cluster funktionieren nur, wenn die IAM-Identity-Center-Identität vollen Tabellenzugriff auf die geschützte Lake-Formation-Tabelle hat. Wenn die Identität keinen vollständigen Tabellenzugriff hat, schlägt die Abfrage fehl.

Verwenden Sie den folgenden Befehl, um dem Benutzer vollen Tabellenzugriff zu gewähren:

```
aws lakeformation grant-permissions --cli-input-json file://grantpermissions.json
json input:
{
  "Principal": {
    "DataLakePrincipalIdentifier": "arn:aws:identitystore:::user/user-id"
  },
  "Resource": {
    "Table": {
      "DatabaseName": "tip_db",
      "Name": "tip_table"
    }
  },
  "Permissions": [
    "ALL"
  ],
  "PermissionsWithGrantOption": [
    "ALL"
  ]
}
```

Mit S3 Access Grants auf einem IAM-Identity-Center-fähigen EMR-Cluster arbeiten

Sie können [S3 Access Grants](#) in Ihren AWS IAM Identity Center aktivierten EMR-Cluster integrieren.

Verwenden Sie S3 Access Grants, um den Zugriff auf Ihre Datensätze von Clustern aus zu autorisieren, die Identity Center verwenden. Erstellen Sie Zuweisungen, um die Berechtigungen zu erweitern, die Sie für IAM-Benutzer, -Gruppen, -Rollen oder für ein Unternehmensverzeichnis

festlegen. Weitere Informationen finden Sie unter [Verwenden von S3 Access Grants mit Amazon EMR](#).

Themen

- [S3-Access-Grants-Instance und -Standort erstellen](#)
- [Grants für Identity-Center-Identitäten erstellen](#)

S3-Access-Grants-Instance und -Standort erstellen

Wenn Sie noch keine haben, erstellen Sie eine S3-Access-Grants-Instance in der AWS-Region , in der Sie Ihren EMR-Cluster starten möchten.

Verwenden Sie den folgenden AWS CLI Befehl, um eine neue Instanz mit dem Namen *MyInstance* zu erstellen:

```
aws s3control-access-grants create-access-grants-instance \  
--account-id 12345678912 \  
--identity-center-arn "identity-center-instance-arn" \  

```

Erstellen Sie dann einen S3-Access-Grants-Standort und ersetzen Sie die roten Werte durch Ihre eigenen:

```
aws s3control-access-grants create-access-grants-location \  
--account-id 12345678912 \  
--location-scope s3:// \  
--iam-role-arn "access-grant-role-arn" \  
--region aa-example-1
```

Note

Definieren Sie den iam-role-arn-Parameter als den accessGrantRole-ARN.

Grants für Identity-Center-Identitäten erstellen

Erstellen Sie abschließend die Grants für die Identitäten, die Zugriff auf Ihren Cluster haben:

```
aws s3control-access-grants create-access-grant \  
--account-id 12345678912 \  

```

```
--access-grants-location-id "default" \  
--access-grants-location-configuration S3SubPrefix="s3-bucket-prefix"  
--permission READ \  
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier="your-identity-center-user-id"
```

Beispielausgabe:

```
{  
  "CreatedAt": "2023-09-21T23:47:24.870000+00:00",  
  "AccessGrantId": "1234-12345-1234-1234567",  
  "AccessGrantArn": "arn:aws:s3:aa-example-1-1:123456789012:access-grants/default/grant/  
xxxx1234-1234-5678-1234-1234567890",  
  "Grantee": {  
    "GranteeType": "DIRECTORY_USER",  
    "GranteeIdentifier": "5678-56789-5678-567890"  
  },  
  "AccessGrantsLocationId": "default",  
  "AccessGrantsLocationConfiguration": {  
    "S3SubPrefix": "myprefix/*"  
  },  
  "Permission": "READ",  
  "GrantScope": "s3://myprefix/*"  
}
```

Überlegungen und Einschränkungen für Amazon EMR mit Identity-Center-Integration

Beachten Sie die folgenden Punkte, wenn Sie IAM Identity Center mit Amazon EMR verwenden:

- Trusted Identity Propagation über Identity Center wird auf Amazon EMR 6.15.0 und höher und nur mit Apache Spark unterstützt. Außerdem wird Trusted Identity Propagation über Identity Center mithilfe der Funktion EMR Runtime Roles auf Amazon EMR 7.8.0 und höher und nur mit Apache Spark unterstützt.
- Um EMR-Cluster mit vertrauenswürdiger Identitätsverbreitung zu aktivieren, müssen Sie die verwenden, AWS CLI um eine Sicherheitskonfiguration zu erstellen, für die vertrauenswürdige Identitätsverbreitung aktiviert ist, und diese Sicherheitskonfiguration verwenden, wenn Sie Ihren Cluster starten. Weitere Informationen finden Sie unter [Eine Identity-Center-fähige Sicherheitskonfiguration erstellen](#).
- Detaillierte Zugriffskontrollen AWS Lake Formation , die Trusted Identity Propagation verwenden, sind für Amazon EMR-Cluster auf EMR-Version 7.2.0 und höher verfügbar. Zwischen den EMR-

Versionen 6.15.0 und 7.1.0 ist nur Zugriffskontrolle auf Tabellenebene verfügbar, die auf AWS Lake Formation basiert.

- Bei Amazon EMR-Clustern, die Trusted Identity Propagation verwenden, gehören SELECT, ALTER TABLE, INSERT INTO und DROP TABLE zu den Vorgängen, die die Zugriffskontrolle auf der Grundlage von Lake Formation mit Apache Spark unterstützen.
- Trusted Identity Propagation mit Amazon EMR wird in den folgenden AWS-Regionen Bereichen unterstützt:
 - `af-south-1` – Afrika (Kapstadt)
 - `ap-east-1` – Asien-Pazifik (Hongkong)
 - `ap-northeast-1` – Asien-Pazifik (Tokio)
 - `ap-northeast-2` – Asien-Pazifik (Seoul)
 - `ap-northeast-3` – Asien-Pazifik (Osaka)
 - `ap-south-1` – Asien-Pazifik (Mumbai)
 - `ap-south-2` – Asien-Pazifik (Hyderabad)
 - `ap-southeast-1` – Asien-Pazifik (Singapur)
 - `ap-southeast-2` – Asien-Pazifik (Sydney)
 - `ap-southeast-3` – Asien-Pazifik (Jakarta)
 - `ap-southeast-4` – Asien-Pazifik (Melbourne)
 - `ca-central-1` – Kanada (Zentral)
 - `eu-central-1` – Europa (Frankfurt)
 - `eu-central-2` – Europa (Zürich)
 - `eu-north-1` – Europa (Stockholm)
 - `eu-south-1` – Europa (Mailand)
 - `eu-south-2` – Europa (Spanien)
 - `eu-west-1` – Europa (Irland)
 - `eu-west-2` – Europa (London)
 - `eu-west-3` – Europa (Paris)
 - `il-central-1` – Israel (Tel Aviv)
 - `me-central-1` – Naher Osten (VAE)
 - `me-south-1` – Naher Osten (Bahrain)
 - `sa-east-1` – Südamerika (São Paulo)

- `us-east-1` – USA Ost (Nord-Virginia)
- `us-east-2` – USA Ost (Ohio)
- `us-west-1` – USA West (Nordkalifornien)
- `us-west-2` – USA West (Oregon)

Integrieren Sie Amazon EMR mit AWS Lake Formation

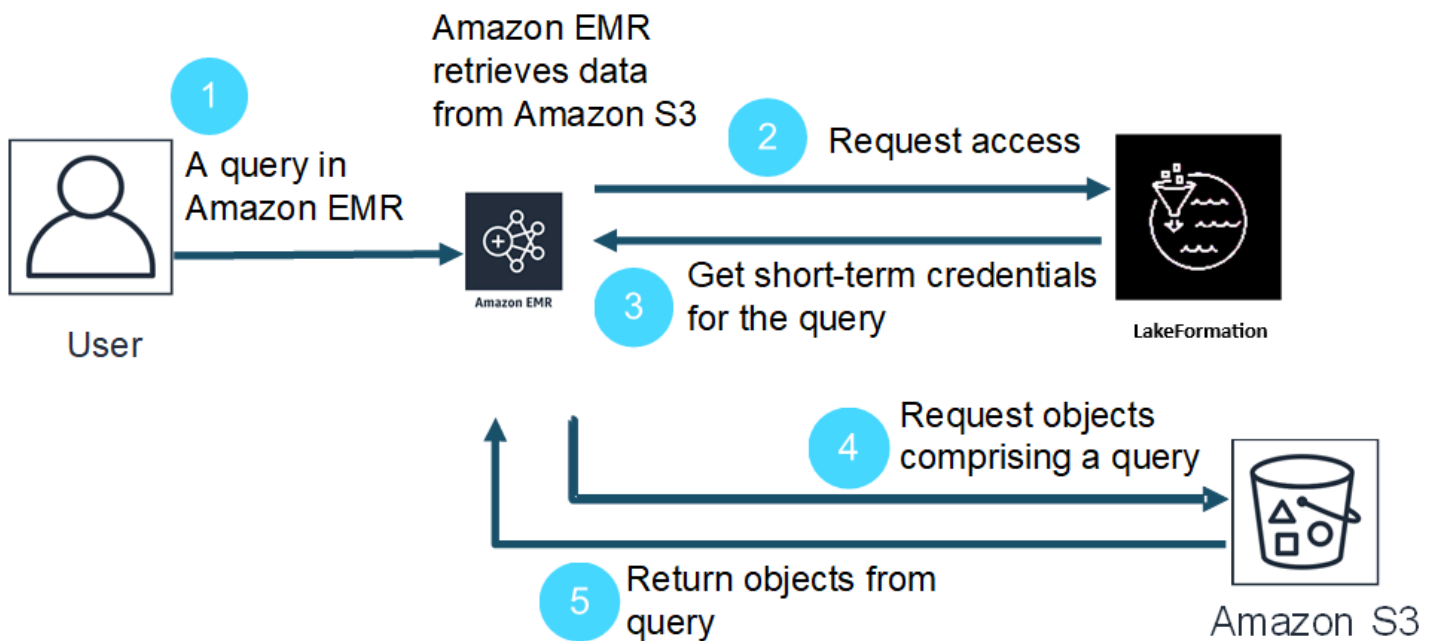
AWS Lake Formation ist ein verwalteter Service, der Sie dabei unterstützt, Daten in einem Amazon Simple Storage Service (S3) Data Lake zu entdecken, zu katalogisieren, zu bereinigen und zu sichern. Lake Formation bietet einen detaillierten Zugriff auf Spaltenebene auf Datenbanken und Tabellen im Glue-Datenkatalog. AWS Weitere Informationen finden Sie unter [Was ist AWS Lake Formation?](#)

Mit Amazon-EMR-Version 6.7.0 und höher können Sie die auf Lake Formation basierende Zugriffskontrolle auf Spark-, Hive- und Presto-Jobs anwenden, die Sie an Amazon-EMR-Cluster senden. Für die Integration mit Lake Formation müssen Sie einen EMR-Cluster mit einer Laufzeit-Rolle erstellen. Eine Laufzeit-Rolle ist eine AWS Identity and Access Management (IAM)-Rolle, der Sie Amazon-EMR-Aufträge oder Abfragen zuordnen. Amazon EMR verwendet diese Rolle dann für den Zugriff auf AWS Ressourcen. Weitere Informationen finden Sie unter [Schritte für Laufzeit-Rollen für Amazon EMR](#).

Wie Amazon EMR mit Lake Formation funktioniert

Nachdem Sie Amazon EMR mit Lake Formation integriert haben, können Sie Abfragen an Amazon EMR-Cluster mit der [StepAPI](#) oder mit SageMaker AI Studio ausführen. Anschließend bietet Lake Formation über temporäre Anmeldeinformationen für Amazon EMR Zugriff auf Daten. Dieser Prozess wird als Anmeldeinformationsvergabe bezeichnet. Weitere Informationen finden Sie unter [Was ist AWS Lake Formation?](#)

Nachfolgend finden Sie einen allgemeinen Überblick darüber, wie Amazon EMR Zugriff auf Daten erhält, die durch Sicherheitsrichtlinien von Lake Formation geschützt sind.



1. Ein Benutzer sendet eine Amazon-EMR-Abfrage für Daten in Lake Formation.
2. Amazon EMR fordert temporäre Anmeldeinformationen von Lake Formation an, um den Benutzerdaten Zugriff zu gewähren.
3. Lake Formation gibt temporäre Anmeldeinformationen zurück.
4. Amazon EMR sendet die Abfrageanfrage zum Abrufen von Daten aus Amazon S3.
5. Amazon EMR empfängt die Daten von Amazon S3, filtert sie und gibt Ergebnisse zurück, die auf den Benutzerberechtigungen basieren, die der Benutzer in Lake Formation definiert hat.

Weitere Informationen zum Hinzufügen von Benutzern und Gruppen zu Lake Formation-Richtlinien finden Sie unter [Erteilen von Datenkatalogberechtigungen](#).

Voraussetzungen

Sie müssen die folgenden Anforderungen erfüllen, bevor Sie Amazon EMR und Lake Formation integrieren können:

- Aktivieren Sie die Laufzeit-Rollenautorisierung in Ihrem Amazon-EMR-Cluster.
- Verwenden Sie den AWS Glue-Datenkatalog als Ihren Metadatenpeicher.
- Definieren und verwalten Sie in Lake Formation Berechtigungen für den Zugriff auf Datenbanken, Tabellen und Spalten im AWS Glue Data Catalog. Weitere Informationen finden Sie unter [Was ist AWS Lake Formation?](#)

Themen

- [Wie Amazon EMR mit Lake Formation funktioniert](#)
- [Apache Hudi und Lake Formation mit Amazon EMR](#)
- [Apache Iceberg und Lake Formation mit Amazon EMR](#)
- [Delta-See und Seenformation auf Amazon EMR](#)
- [Überlegungen zu Amazon EMR mit Lake Formation](#)
- [Verwenden einer Lake Formation Formation-Job-Laufzeitrolle mit vollständigem Tabellenzugriff](#)

Wie Amazon EMR mit Lake Formation funktioniert

Wenn Sie mit Amazon EMR 6.15.0 und höher Spark-Jobs auf Amazon EMR auf EC2 Clustern ausführen, die auf Daten im AWS Glue-Datenkatalog zugreifen, können Sie damit Berechtigungen auf Tabellen-, Zeilen-, Spalten- und Zellenebene auf Hudi-, Iceberg- oder Delta Lake-basierte Tabellen anwenden. AWS Lake Formation

In diesem Abschnitt erfahren Sie, wie Sie eine Sicherheitskonfiguration erstellen und Lake Formation so einrichten, dass es mit Amazon EMR funktioniert. Wir gehen auch darauf ein, wie Sie einen Cluster mit der Sicherheitskonfiguration starten, die Sie für Lake Formation erstellt haben.

Schritt 1: Eine Laufzeit-Rolle für Ihren EMR-Cluster einrichten

Um die Laufzeit-Rolle für Ihren Cluster einzurichten, erstellen Sie eine Sicherheitskonfiguration. Mit einer Sicherheitskonfiguration können Sie konsistente Sicherheits-, Autorisierungs- und Authentifizierungsoptionen für alle Ihre Cluster anwenden.

1. Erstellen Sie eine Datei mit dem Namen `lf-runtime-roles-sec-cfg.json` und den folgenden Inhalten.

```
{
  "AuthorizationConfiguration": {
    "IAMConfiguration": {
      "EnableApplicationScopedIAMRole": true,
      "ApplicationScopedIAMRoleConfiguration": {
        "PropagateSourceIdentity": true
      }
    },
    "LakeFormationConfiguration": {
```



```

        "AuthorizedSessionTagValue": "Amazon EMR"
    },
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": true,
        "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {<certificate-configuration>}
        }
    }
}

```

2. Um als Nächstes sicherzustellen, dass das Sitzungs-Tag Lake Formation autorisieren kann, setzen Sie die LakeFormationConfiguration/AuthorizedSessionTagValue-Eigenschaft auf Amazon EMR.
3. Verwenden Sie den folgenden Befehl, um die Amazon-EMR-Sicherheitskonfiguration zu erstellen.

```

aws emr create-security-configuration \
--name 'iamconfig-with-iam-lf' \
--security-configuration file://lf-runtime-roles-sec-cfg.json

```

Alternativ können Sie die [Amazon-EMR-Konsole](#) verwenden, um eine Sicherheitskonfiguration mit benutzerdefinierten Einstellungen zu erstellen.

Schritt 2: Starten eines Amazon-EMR-Clusters

Jetzt sind Sie bereit, einen EMR-Cluster mit der Sicherheitskonfiguration zu starten, die Sie im vorherigen Schritt erstellt haben. Weitere Informationen zum Erstellen einer Sicherheitskonfiguration finden Sie unter [Verwenden Sie Sicherheitskonfigurationen, um die Amazon EMR-Cluster-Sicherheit einzurichten](#) und [Schritte für Laufzeit-Rollen für Amazon EMR](#).

Schritt 3a: Lake-Formation-basierte Berechtigungen auf Tabellenebene mit Amazon-EMR-Laufzeitrollen einrichten

Wenn Sie keine differenzierte Zugriffskontrolle auf Spalten-, Zeilen- oder Zellenebene benötigen, können Sie mit Glue-Datenkatalog Berechtigungen auf Tabellenebene einrichten. Um den Zugriff auf Tabellenebene zu aktivieren, navigieren Sie zur AWS Lake Formation Konsole und wählen Sie im Bereich Administration in der Seitenleiste die Option Anwendungsintegrationseinstellungen aus. Aktivieren Sie dann die folgende Option und wählen Sie Save (Speichern) aus:

Allow external engines to access data in Amazon S3 locations with full table access (Externen Engines den Zugriff auf Daten an Amazon-S3-Standorten mit vollständigem Tabellenzugriff erlauben)

[AWS Lake Formation](#) > Application integration settings

Application integration settings [Learn more](#)

Application integration settings

Use the options below to control which third-party engines are allowed to read and filter data in Amazon S3 locations registered with Lake Formation.

☐ Allow external engines to filter data in Amazon S3 locations registered with Lake Formation

Check this box to allow third-party engines to access data in Amazon S3 locations that are registered with Lake Formation.

☒ Allow external engines to access data in Amazon S3 locations with full table access

When you enable this option, Lake Formation will return credentials to the integrated application directly without IAM session tag validation.

Cancel **Save**

Schritt 3b: Lake-Formation-basierte Berechtigungen auf Spalten-, Zeilen- oder Zellenebene mit Amazon-EMR-Laufzeitrollen einrichten

Um Berechtigungen auf Tabellen- und Spaltenebene mit Lake Formation anzuwenden, muss der Data-Lake-Administrator für Lake Formation Amazon EMR als Wert für die Sitzungs-Tag-Konfiguration `AuthorizedSessionTagValue` festlegen. Lake Formation verwendet dieses Sitzungs-Tag, um Anrufer zu autorisieren und Zugriff auf den Data Lake zu gewähren. Sie können dieses Sitzungs-Tag im Abschnitt Externe Datenfilterung der Lake-Formation-Konsole festlegen. Ersetze es **123456789012** durch deine eigene ID. AWS-Konto

[Lake Formation](#) > External data filtering

External data filtering

External data filtering settings

Use the options below to control which third-party engines are allowed to read and filter data in Amazon S3 locations registered with Lake Formation.

☒ **Allow external engines to filter data in Amazon S3 locations registered with Lake Formation**
Check this box to allow third-party engines to access data in Amazon S3 locations that are registered with Lake Formation.

Session tag values
Enter one or more strings that match the LakeFormationAuthorizedCaller session tag defined for third-party engines.

Clear all

Amazon EMR ✕

Enter one or several string values separated by comma.

account IDs
Enter the external account IDs from where third-party engines are allowed to access locations registered with Lake Formation.

Clear all

123456789012 ✕
Account

Enter one or more account IDs. Press enter after each ID.

Cancel **Save**

Schritt 4: AWS Glue- und Lake Formation Formation-Grants für Amazon EMR- Runtime-Rollen konfigurieren

Um mit der Einrichtung der auf Lake Formation basierenden Zugriffskontrolle mit Amazon EMR- Runtime-Rollen fortzufahren, müssen Sie AWS Glue- und Lake Formation Formation-Grants für Amazon EMR- Runtime-Rollen konfigurieren. Damit Ihre IAM-Laufzeitrollen mit Lake Formation interagieren können, gewähren Sie ihnen Zugriff mit `lakeformation:GetDataAccess` und `glue:Get*`.

Lake Formation Formations-Berechtigungen kontrollieren den Zugriff auf AWS Glue Data Catalog-Ressourcen, Amazon S3 S3-Standorte und die zugrunde liegenden Daten an diesen Standorten. IAM-Berechtigungen kontrollieren den Zugriff auf Lake Formation und AWS Glue APIs sowie auf Ressourcen. Obwohl Sie möglicherweise über die Lake-Formation-Berechtigung verfügen, auf eine Tabelle im Datenkatalog (SELECT) zuzugreifen, schlägt Ihr Vorgang fehl, wenn Sie nicht über die IAM-Berechtigung für die `glue:Get*`-API verfügen. Weitere Informationen zur Zugriffskontrolle für Lake Formation finden Sie unter [Übersicht über die Zugriffskontrolle für Lake Formation](#).

1. Erstellen Sie die Datei `emr-runtime-roles-lake-formation-policy.json` mit folgendem Inhalt.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "LakeFormationManagedAccess",
    "Effect": "Allow",
    "Action": [
      "lakeformation:GetDataAccess",
      "glue:Get*",
      "glue:Create*",
      "glue:Update*"
    ],
    "Resource": "*"
  }
}
```

2. Erstellen Sie die zugehörige IAM-Richtlinie.

```
aws iam create-policy \
--policy-name emr-runtime-roles-lake-formation-policy \
--policy-document file:///emr-runtime-roles-lake-formation-policy.json
```

3. Um diese Richtlinie Ihren IAM-Laufzeit-Rollen zuzuweisen, folgen Sie den Schritten unter [Verwalten von AWS Lake Formation -Berechtigungen](#).

Sie können jetzt Laufzeit-Rollen und Lake Formation verwenden, um Berechtigungen auf Tabellen- und Spaltenebene anzuwenden. Sie können auch eine Quellidentität verwenden, um Aktionen zu steuern und Vorgänge zu AWS CloudTrailüberwachen. Ein detailliertes end-to-end Beispiel finden Sie unter [Schritte zur Einführung von Runtime-Rollen für Amazon EMR](#).

Informationen zur Integration mit Iceberg und AWS Glue Data Catalog für eine Hierarchie mit mehreren Katalogen finden Sie unter [Konfigurieren von Spark für den Zugriff auf eine Hierarchie mit mehreren Katalogen in AWS Glue Data Catalog](#).

Apache Hudi und Lake Formation mit Amazon EMR

Die Amazon EMR-Versionen 6.15.0 und höher bieten Unterstützung für eine differenzierte Zugriffskontrolle, die auf Apache Hudi basiert, wenn Sie Daten AWS Lake Formation mit Spark SQL lesen und schreiben. Amazon EMR unterstützt die Zugriffskontrolle auf Tabellen-, Zeilen-, Spalten- und Zellenebene mit Apache Hudi. Mit dieser Funktion können Sie Snapshot-Abfragen für copy-on-write Tabellen ausführen, um den neuesten Snapshot der Tabelle zu einem bestimmten Commit- oder Komprimierungszeitpunkt abzufragen.

Derzeit muss ein Lake Formation-fähiger Amazon EMR-Cluster die Commit-Zeitspalte von Hudi abrufen, um inkrementelle Abfragen und Zeitreiseabfragen durchzuführen. Die Syntax und die Funktion von Spark werden nicht unterstützt. `timestamp as of Spark.read()` Die richtige Syntax ist `select * from table where _hoodie_commit_time <= point_in_time`. Weitere Informationen finden Sie unter [Point-in-Time-Time-Travel-Abfragen in der Hudi-Tabelle](#).

Die folgende Unterstützungsmatrix listet einige Kernfeatures von Apache Hudi mit Lake Formation auf:

	Kopieren Sie beim Schreiben	Beim Lesen zusammenführen (MoR)
Snapshot-Abfragen – Spark SQL	✓	✓
Leseoptimierte Abfragen – Spark SQL	✓	✓
Inkrementelle Abfragen	✓	✓
Zeitreiseabfragen	✓	✓
Metadaten-Tabellen	✓	✓
DML- INSERT -Befehle	✓	✓
DDL-Befehle		
Spark-Datenquellenabfragen		

	Kopieren Sie beim Schreiben	Beim Lesen zusammenführen (MoR)

Spark-Datenquellenschreibvorgänge

Abfragen von Hudi-Tabellen

In diesem Abschnitt wird gezeigt, wie Sie die oben beschriebenen unterstützten Abfragen auf einem Lake-Formation-fähigen Cluster ausführen können. Bei der Tabelle sollte es sich um eine registrierte Katalogtabelle handeln.

1. Verwenden Sie die folgenden Befehle, um die Spark-Shell zu starten.

```
spark-sql
--jars /usr/lib/hudi/hudi-spark-bundle.jar \
--conf spark.serializer=org.apache.spark.serializer.KryoSerializer \
--conf
spark.sql.catalog.spark_catalog=org.apache.spark.sql.hudi.catalog.HoodieCatalog \
--conf
spark.sql.extensions=org.apache.spark.sql.hudi.HoodieSparkSessionExtension,com.amazonaws.em
\
--conf spark.sql.catalog.spark_catalog.lf.managed=true
```

Wenn Sie möchten, dass Lake Formation einen Datensatzserver zur Verwaltung Ihres Spark-Katalogs verwendet, legen Sie

`spark.sql.catalog.<managed_catalog_name>.lf.managed` den Wert auf `true` fest.

2. Verwenden Sie die folgenden Befehle, um den neuesten Snapshot der copy-on-write Tabellen abzufragen.

```
SELECT * FROM my_hudi_cow_table
```

```
spark.read.table("my_hudi_cow_table")
```

3. Um die neuesten komprimierten Daten von MOR-Tabellen abzufragen, können Sie die leseoptimierte Tabelle mit dem Suffix `_ro` abfragen:

```
SELECT * FROM my_hudi_mor_table_ro
```

```
spark.read.table("my_hudi_mor_table_ro")
```

Note

Die Leistung von Lesevorgängen auf Lake Formation-Clustern kann aufgrund von Optimierungen, die nicht unterstützt werden, langsamer sein. Zu diesen Features gehören das Auflisten von Dateien auf der Grundlage von Hudi-Metadaten und das Überspringen von Daten. Wir empfehlen Ihnen, die Leistung Ihrer Anwendung zu testen, um sicherzustellen, dass sie Ihrem SLA entspricht.

Apache Iceberg und Lake Formation mit Amazon EMR

Die Amazon EMR-Versionen 6.15.0 und höher bieten Unterstützung für eine differenzierte Zugriffskontrolle, die auf Apache Iceberg basiert, wenn Sie Daten AWS Lake Formation mit Spark SQL lesen und schreiben. Amazon EMR unterstützt die Zugriffskontrolle auf Tabellen-, Zeilen-, Spalten- und Zellenebene mit Apache Iceberg. Mit dieser Funktion können Sie Snapshot-Abfragen für copy-on-write Tabellen ausführen, um den neuesten Snapshot der Tabelle zu einem bestimmten Commit- oder Komprimierungszeitpunkt abzufragen.

Wenn Sie das Iceberg-Format verwenden möchten, legen Sie die folgenden Konfigurationen fest. Ersetzen Sie **DB_LOCATION** durch den Amazon-S3-Pfad, in dem sich Ihre Iceberg-Tabellen befinden, und ersetzen Sie die Platzhalter für Region und Konto-ID durch Ihre eigenen Werte.

```
spark-sql \  
--conf  
  spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions,com.ama  
  
--conf spark.sql.catalog.iceberg_catalog=org.apache.iceberg.spark.SparkCatalog  
--conf spark.sql.catalog.iceberg_catalog.warehouse=s3://DB_LOCATION  
--conf spark.sql.catalog.iceberg_catalog.catalog-  
impl=org.apache.iceberg.aws.glue.GlueCatalog  
--conf spark.sql.catalog.iceberg_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO  
--conf spark.sql.catalog.iceberg_catalog.glue.account-id=ACCOUNT_ID  
--conf spark.sql.catalog.iceberg_catalog.glue.id=ACCOUNT_ID  
--conf spark.sql.catalog.iceberg_catalog.client.assume-role.region=AWS_REGION  
--conf spark.sql.secureCatalog=iceberg_catalog
```

Wenn Sie möchten, dass Lake Formation einen Datensatzserver zur Verwaltung Ihres Spark-Katalogs verwendet, legen Sie `spark.sql.catalog.<managed_catalog_name>.lf.managed` den Wert auf `true` fest.

Sie sollten auch darauf achten, die folgenden Rollenübernahme-Einstellungen NICHT weiterzugeben:

```
--conf spark.sql.catalog.my_catalog.client.assume-role.region
--conf spark.sql.catalog.my_catalog.client.assume-role.arn
--conf spark.sql.catalog.my_catalog.client.assume-
role.tags.LakeFormationAuthorizedCaller
```

Die folgende Unterstützungsmatrix listet einige Kernfeatures von Apache Iceberg mit Lake Formation auf:

	Kopieren Sie beim Schreiben	Beim Lesen zusammenführen (MoR)
Snapshot-Abfragen – Spark SQL	✓	✓
Leseoptimierte Abfragen – Spark SQL	✓	✓
Inkrementelle Abfragen	✓	✓
Zeitreiseabfragen	✓	✓
Metadaten-Tabellen	✓	✓
DML- INSERT -Befehle	✓	✓
DDL-Befehle		
Spark-Datenquellenabfragen		
Spark-Datenquellenschreibvorgänge		

Delta-See und Seenformation auf Amazon EMR

Die Amazon EMR-Versionen 6.15.0 und höher bieten Unterstützung für eine differenzierte Zugriffskontrolle, die auf Delta Lake basiert, wenn Sie Daten AWS Lake Formation mit Spark SQL

lesen und schreiben. Amazon EMR unterstützt die Zugriffskontrolle auf Tabellen-, Zeilen-, Spalten- und Zellenebene mit Delta Lake. Mit dieser Funktion können Sie Snapshot-Abfragen für copy-on-write Tabellen ausführen, um den neuesten Snapshot der Tabelle zu einem bestimmten Commit- oder Komprimierungszeitpunkt abzufragen.

Führen Sie den folgenden Befehl aus, um Delta Lake mit Lake Formation zu verwenden.

```
spark-sql \  
--conf  
  spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension,com.amazonaws.emr.recordserver.co  
 \  
--conf spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog  
 \  
--conf spark.sql.catalog.spark_catalog.lf.managed=true
```

Wenn Sie möchten, dass Lake Formation einen Datensatzserver zur Verwaltung Ihres Spark-Katalogs verwendet, legen Sie `spark.sql.catalog.<managed_catalog_name>.lf.managed` den Wert auf `true` fest.

Die folgende Unterstützungsmatrix listet einige Kernfeatures von Delta Lake mit Lake Formation auf:

	Kopieren Sie beim Schreiben	Beim Lesen zusammenführen (MoR)
Snapshot-Abfragen – Spark SQL	✓	✓
Leseoptimierte Abfragen – Spark SQL	✓	✓
Inkrementelle Abfragen	Nicht unterstützt	Nicht unterstützt
Zeitreiseabfragen	Nicht unterstützt	Nicht unterstützt
Metadaten-Tabellen	✓	✓
DML- INSERT -Befehle	✓	✓
DDL-Befehle		
Spark-Datenquellenabfragen		
Spark-Datenquellenschreibvorgänge		

Erstellen einer Delta Lake-Tabelle im AWS Glue Data Catalog

Amazon EMR mit Lake Formation unterstützt keine DDL-Befehle und die Erstellung von Delta-Tabellen. Gehen Sie wie folgt vor, um Tabellen im AWS Glue-Datenkatalog zu erstellen.

1. Verwenden Sie das folgende Beispiel, um eine Delta-Tabelle zu erstellen. Stellen Sie sicher, dass Ihr S3-Standort existiert.

```
spark-sql \  
--conf "spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension" \  
--conf  
  "spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog"  
  
> CREATE DATABASE if not exists <DATABASE_NAME> LOCATION 's3://<S3_LOCATION>/  
transactionaldata/native-delta/<DATABASE_NAME>/';  
> CREATE TABLE <TABLE_NAME> (x INT, y STRING, z STRING) USING delta;  
> INSERT INTO <TABLE_NAME> VALUES (1, 'a1', 'b1');
```

2. Um die Details Ihrer Tabelle zu sehen, gehen Sie zu <https://console.aws.amazon.com/glue/>.
3. Erweitern Sie in der linken Navigationsleiste den Datenkatalog, wählen Sie Tabellen und dann die Tabelle aus, die Sie erstellt haben. Unter Schema sollten Sie sehen, dass die Delta-Tabelle, die Sie mit Spark erstellt haben, alle Spalten in einem Datentyp von array<string> in AWS Glue speichert.
4. Um Filter auf Spalten- und Zellenebene in Lake Formation zu definieren, entfernen Sie die col1 Spalte aus Ihrem Schema und fügen Sie dann die Spalten hinzu, die sich in Ihrem Tabellenschema befinden. Fügen Sie in diesem Beispiel die Spalten xy, und hinzu. z

Überlegungen zu Amazon EMR mit Lake Formation

Beachten Sie Folgendes, wenn Sie Amazon EMR mit AWS Lake Formation verwenden.

- [Zugriffskontrolle auf Tabellenebene](#) ist auf Clustern mit Amazon-EMR-Versionen 6.13 und höher verfügbar.
- [Differenzierte Zugriffskontrolle](#) auf Zeilen-, Spalten- und Zellenebene ist auf Clustern mit Amazon-EMR-Versionen 6.15 und höher verfügbar.
- Benutzer mit Zugriff auf eine Tabelle können auf alle Eigenschaften dieser Tabelle zugreifen. Wenn Sie eine auf Lake Formation basierende Zugriffskontrolle für eine Tabelle haben, überprüfen

Sie die Tabelle, um sicherzustellen, dass die Eigenschaften keine vertraulichen Daten oder Informationen enthalten.

- Amazon-EMR-Cluster mit Lake Formation unterstützen den Fallback von Spark auf HDFS nicht, wenn Spark Tabellenstatistiken sammelt. Dies trägt normalerweise zur Optimierung der Abfrageleistung bei.
- Zu den Vorgängen, die Zugriffskontrollen auf der Grundlage von Lake Formation mit nicht verwalteten Apache-Spark-Tabellen unterstützen, gehören `INSERT INTO` und `INSERT OVERWRITE`.
- Zu den Vorgängen, die auf Lake Formation mit Apache Spark und Apache Hive basierende Zugriffskontrollen unterstützen `SELECT`, `DESCRIBE`, `SHOW DATABASE`, `SHOW TABLE`, `SHOW COLUMN` und `SHOW PARTITION`.
- Amazon EMR unterstützt keine Zugriffskontrolle auf die folgenden auf Lake Formation basierenden Vorgänge:
 - Schreibt in geregelte Tabellen
 - Amazon EMR unterstützt `CREATE TABLE` nicht. Amazon EMR 6.10.0 und höher unterstützt `ALTER TABLE`.
 - Andere DML-Anweisungen als `INSERT`-Befehle.
- Es gibt Leistungsunterschiede zwischen derselben Abfrage mit und ohne Lake-Formation-basierte Zugriffskontrolle.
- Sie können Amazon EMR nur mit Lake Formation für Spark-Jobs verwenden.
- Die Weitergabe vertrauenswürdiger Identitäten wird bei einer Hierarchie mit mehreren Katalogen in Glue Data Catalog nicht unterstützt. Weitere Informationen finden Sie unter [Arbeiten mit einer Hierarchie mit mehreren Katalogen im AWS Glue-Datenkatalog](#).

Verwenden einer Lake Formation Formation-Job-Laufzeitrolle mit vollständigem Tabellenzugriff

Die Themen in diesem Abschnitt behandeln, wie Sie über EMR Spark-Jobs oder interaktive Sitzungen mit vollständigem Tabellenzugriff auf Lake Formation Formation-geschützte Glue Data-Katalogtabellen zugreifen können.

Lake Formation ungefilterter Zugang für EMR am EC2

Mit den Amazon EMR-Versionen 7.8.0 und höher können Sie AWS Lake Formation mit Glue Data Catalog nutzen, wobei die Job-Runtime-Rolle über vollständige Tabellenberechtigungen verfügt.

Diese Funktion ermöglicht Ihnen das Lesen und Schreiben in Tabellen, die durch Lake Formation geschützt sind, von Ihren EMR Spark-Batch- und interaktiven Jobs aus.

Lake Formation mit vollem Tabellenzugriff verwenden

Sie können über EMR Spark-Jobs oder interaktive Sitzungen, in denen die Runtime-Rolle des Jobs vollen Tabellenzugriff hat, auf AWS Lake Formation Formation-geschützte Glue Data-Katalogtabellen zugreifen. Sie müssen AWS Lake Formation auf dem EMR-Cluster nicht aktivieren. Wenn ein Spark-Job für Full Table Access (FTA) konfiguriert ist, werden AWS Lake Formation-Anmeldeinformationen zum Lesen/Schreiben von S3-Daten für in AWS Lake Formation registrierte Tabellen verwendet, während die Anmeldeinformationen für die Runtime-Rolle des Jobs für read/write Tabellen verwendet werden, die nicht bei AWS Lake Formation registriert sind.

Note

Eine feinkörnige Zugriffskontrolle von Lake Formation ist für einen Job nicht erforderlich oder wird nicht unterstützt, um Full Table Access (FTA) auszuführen.

Note

Für den vollständigen Tabellenzugriff ist keine EMR-Laufzeitrolle erforderlich, und er funktioniert sowohl mit der Runtime- als auch mit der Service-Rolle für EC2 Cluster-Instances. Weitere Informationen finden Sie unter [Schritte zur Servicerolle für EC2 Cluster-Instances \(EC2 Instance-Profil\)](#) oder [Runtime-Rollen für Amazon EMR](#).

Schritt 1: Vollständigen Tabellenzugriff in Lake Formation aktivieren

Um den Modus Full Table Access (FTA) zu verwenden, müssen Sie Drittanbieter-Abfrage-Engines den Zugriff auf Daten ohne die IAM-Sitzungs-Tag-Validierung in AWS Lake Formation ermöglichen. Folgen Sie zur Aktivierung den Schritten unter [Anwendungsintegration für vollständigen Tabellenzugriff](#).

Note

Beim Zugriff auf kontenübergreifende Tabellen muss der vollständige Tabellenzugriff sowohl im Produzenten- als auch im Verbraucherkonto aktiviert sein. Ebenso muss diese

Einstellung beim Zugriff auf regionsübergreifende Tabellen sowohl in Erzeuger- als auch in Verbraucherregionen aktiviert sein.

Schritt 2: Richten Sie die IAM-Berechtigungen für die Job-Runtime-Rolle ein

Für den Lese- oder Schreibzugriff auf zugrunde liegende Daten benötigt eine Job-Runtime-Rolle zusätzlich zu den Lake Formation Formation-Berechtigungen die `lakeformation:GetDataAccess` IAM-Berechtigung. Mit dieser Berechtigung gewährt Lake Formation die Anforderung von temporären Anmeldeinformationen für den Zugriff auf die Daten.

Im Folgenden finden Sie eine Beispielrichtlinie für die Bereitstellung von IAM-Berechtigungen für den Zugriff auf ein Skript in Amazon S3, für das Hochladen von Protokollen auf S3, für AWS Glue-API-Berechtigungen und für den Zugriff auf Lake Formation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ScriptAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3::*.amzn-s3-demo-bucket/scripts"
      ]
    },
    {
      "Sid": "LoggingAccess",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/logs/*"
      ]
    },
    {
      "Sid": "GlueCatalogAccess",
      "Effect": "Allow",
      "Action": [
```

```

        "glue:Get*",
        "glue:Create*",
        "glue:Update*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Sid": "LakeFormationAccess",
    "Effect": "Allow",
    "Action": [
        "lakeformation:GetDataAccess"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

Schritt 2.1 Lake Formation Formation-Berechtigungen konfigurieren

- Spark-Jobs, die Daten aus S3 lesen, benötigen die Lake Formation SELECT-Berechtigung.
- Spark-Jobs, für die write/delete Daten in S3 ist die Lake Formation ALL (SUPER) -Genehmigung erforderlich.
- Spark-Jobs, die mit dem Glue-Datenkatalog interagieren, benötigen die entsprechenden DESCRIBE-, ALTER- und DROP-Berechtigungen.

Weitere Informationen finden Sie unter [Erteilen von Berechtigungen für Datenkatalogressourcen](#).

Schritt 3: Initialisieren Sie eine Spark-Sitzung für den vollständigen Tabellenzugriff mit Lake Formation

Voraussetzungen

AWS Der Glue Data Catalog muss als Metastore konfiguriert werden, um auf Lake Formation-Tabellen zugreifen zu können.

Legen Sie die folgenden Einstellungen fest, um den Glue-Katalog als Metastore zu konfigurieren:

```
--conf spark.sql.catalogImplementation=hive
```

```
--conf
spark.hive.metastore.client.factory.class=com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalog
```

Informationen zur Aktivierung von Data Catalog for EMR finden Sie unter [Verwenden des AWS Glue-Datenkatalogs mit Spark auf Amazon EMR](#). EC2

Zugreifen auf Tabellen

Um auf Tabellen zuzugreifen, die bei AWS Lake Formation registriert sind, müssen die folgenden Konfigurationen während der Spark-Initialisierung festgelegt werden, um Spark für die Verwendung von AWS Lake Formation Anmeldeinformationen zu konfigurieren.

Hive

```
--conf
spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.glue.accesscontrol.AWSLakeFormation
--conf spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true
--conf spark.hadoop.fs.s3.folderObject.autoAction.disabled=true
--conf spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true
--conf spark.sql.catalog.createDirectoryAfterTable.enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
```

Iceberg

```
--conf spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkSessionCatalog
--conf spark.sql.catalog.spark_catalog.warehouse=S3_DATA_LOCATION
--conf spark.sql.catalog.spark_catalog.client.region=REGION
--conf spark.sql.catalog.spark_catalog.type=glue
--conf spark.sql.catalog.spark_catalog.glue.account-id=ACCOUNT_ID
--conf spark.sql.catalog.spark_catalog.glue.lakeformation-enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
```

- `spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.glue.accesscontrol.AWSLakeFormation`
Konfigurieren Sie das EMR-Dateisystem (EMRFS) so, dass es AWS Lake Formation S3-Anmeldeinformationen für registrierte Lake Formation-Tabellen verwendet. Wenn die Tabelle nicht registriert ist, verwenden Sie die Anmeldeinformationen für die Runtime-Rolle des Jobs.
- `spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true` und `spark.hadoop.fs.s3.folderObject.autoAction.disabled=true`
Konfigurieren Sie EMRFS so, dass beim Erstellen von S3-Ordnern der Inhaltstyp-Header `application/x-directory` anstelle des `$folder$` Suffixes verwendet wird. Dies ist beim Lesen von

Lake Formation-Tabellen erforderlich, da die Anmeldeinformationen von Lake Formation das Lesen von Tabellenordnern mit dem Suffix `$folder$` nicht zulassen.

- `spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true`: Konfigurieren Sie Spark so, dass die Überprüfung, ob der Tabellenspeicherort leer ist, vor der Erstellung übersprungen wird. Dies ist für in Lake Formation registrierte Tabellen erforderlich, da die Anmeldeinformationen für Lake Formation zur Überprüfung des leeren Speicherorts erst nach der Erstellung der Glue Data Catalog-Tabelle verfügbar sind. Ohne diese Konfiguration validieren die Anmeldeinformationen für die Runtime-Rolle des Jobs den Speicherort der leeren Tabelle.
- `spark.sql.catalog.createDirectoryAfterTable.enabled=true`: Konfigurieren Sie Spark so, dass der Amazon S3 S3-Ordner nach der Tabellenerstellung im Hive-Metastore erstellt wird. Dies ist für in Lake Formation registrierte Tabellen erforderlich, da die Anmeldeinformationen für Lake Formation zum Erstellen des S3-Ordners erst nach der Erstellung der Glue Data Catalog-Tabelle verfügbar sind.
- `spark.sql.catalog.dropDirectoryBeforeTable.enabled=true`: Konfigurieren Sie Spark so, dass der S3-Ordner vor dem Löschen der Tabelle im Hive-Metastore gelöscht wird. Dies ist für in Lake Formation registrierte Tabellen erforderlich, da die Lake Formation Anmeldeinformationen zum Löschen des S3-Ordners nach dem Löschen der Tabelle aus dem Glue-Datenkatalog nicht verfügbar sind.
- `spark.sql.catalog.<catalog>.glue.lakeformation-enabled=true`: Konfigurieren Sie den Iceberg-Katalog so, dass er AWS Lake Formation S3-Anmeldeinformationen für registrierte Lake Formation-Tabellen verwendet. Wenn die Tabelle nicht registriert ist, verwenden Sie die Standardanmeldedaten für die Umgebung.

Konfigurieren Sie den vollständigen Tabellenzugriffsmodus in SageMaker Unified Studio

Um über interaktive Spark-Sitzungen in JupyterLab Notebooks auf registrierte Tabellen von Lake Formation zuzugreifen, müssen Sie den Kompatibilitätsberechtigungsmodus verwenden. Verwenden Sie den `%%configure` magischen Befehl, um Ihre Spark-Konfiguration einzurichten. Wählen Sie die Konfiguration basierend auf Ihrem Tabellentyp aus:

For Hive tables

```
%%configure -f
{
  "conf": {
    "spark.hadoop.fs.s3.credentialsResolverClass":
    "com.amazonaws.glue.accesscontrol.AWSLakeFormationCredentialResolver",
```



```

    "spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject": true,
    "spark.hadoop.fs.s3.folderObject.autoAction.disabled": true,
    "spark.sql.catalog.skipLocationValidationOnCreateTable.enabled": true,
    "spark.sql.catalog.createDirectoryAfterTable.enabled": true,
    "spark.sql.catalog.dropDirectoryBeforeTable.enabled": true
  }
}

```

For Iceberg tables

```

%%configure -f
{
  "conf": {
    "spark.sql.catalog.spark_catalog":
"org.apache.iceberg.spark.SparkSessionCatalog",
    "spark.sql.catalog.spark_catalog.warehouse": "S3_DATA_LOCATION",
    "spark.sql.catalog.spark_catalog.client.region": "REGION",
    "conf spark.sql.catalog.spark_catalog.type": "glue",
    "spark.sql.catalog.spark_catalog.glue.account-id": "ACCOUNT_ID",
    "spark.sql.catalog.spark_catalog.glue.lakeformation-enabled": true,
    "spark.sql.catalog.dropDirectoryBeforeTable.enabled": true
  }
}

```

Ersetzen Sie die Platzhalter:

- *S3_DATA_LOCATION*: Ihr S3-Bucket-Pfad
- *REGION*: AWS Region (z. B. us-east-1)
- *ACCOUNT_ID*: Ihre Konto-ID AWS

Note

Sie müssen diese Konfigurationen festlegen, bevor Sie Spark-Operationen in Ihrem Notebook ausführen.

Unterstützte Vorgänge

Bei diesen Vorgängen werden AWS Lake Formation Formation-Anmeldeinformationen für den Zugriff auf die Tabellendaten verwendet.

- CREATE TABLE
- ALTER TABLE
- INSERT INTO
- INSERT OVERWRITE
- UPDATE
- MERGE INTO
- DELETE FROM
- TABELLE ANALYSIEREN
- TABELLE REPARIEREN
- DROP TABLE
- Spark-Datenquellenabfragen
- Spark-Datenquellenschreibvorgänge

Note

Operationen, die oben nicht aufgeführt sind, verwenden weiterhin IAM-Berechtigungen für den Zugriff auf Tabellendaten.

Überlegungen

- Wenn eine Hive-Tabelle mit einem Job erstellt wird, für den der vollständige Tabellenzugriff nicht aktiviert ist, und keine Datensätze eingefügt werden, schlagen nachfolgende Lese- oder Schreibvorgänge aus einem Job mit vollständigem Tabellenzugriff fehl. Dies liegt daran, dass EMR Spark ohne vollständigen Tabellenzugriff das `$folder$` Suffix zum Namen des Tabellenordners hinzufügt. Um dieses Problem zu lösen, können Sie entweder:
 - Fügen Sie mindestens eine Zeile aus einem Job in die Tabelle ein, für den FTA nicht aktiviert ist.
 - Konfigurieren Sie den Job, für den FTA nicht aktiviert ist, so, dass er in S3 kein `$folder$` Suffix im Ordernamen verwendet. Dies kann durch die Einstellung der Spark-Konfiguration erreicht werden. `spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true`

- Erstellen Sie mit der S3-Konsole oder AWS der S3-CLI einen AWS S3-Ordner am Speicherort `s3://path/to/table/table_name` der Tabelle.
- Full Table Access funktioniert ausschließlich mit dem EMR-Dateisystem (EMRFS). Das S3A-Dateisystem ist nicht kompatibel.
- Full Table Access wird für Hive- und Iceberg-Tabellen unterstützt. Support für Hudi- und Delta-Tabellen wurde noch nicht hinzugefügt.
- Jobs, die auf Tabellen mit Lake Formation Fine-Grained Access Control (FGAC) -Regeln oder Glue Data Catalog Views verweisen, schlagen fehl.
- Der vollständige Tabellenzugriff unterstützt Spark Streaming nicht.
- Beim Schreiben von Spark DataFrame in eine Lake Formation-Tabelle wird nur der APPEND-Modus unterstützt: `df.write.mode("append").saveAsTable(table_name)`

Integrieren Sie Amazon EMR mit Apache Ranger

Mit Amazon EMR 5.32.0 können Sie einen Cluster starten, der nativ in Apache Ranger integriert ist. Apache Ranger ist ein Open-Source-Framework zur Aktivierung, Überwachung und Verwaltung einer umfassenden Datensicherheit auf der gesamten Hadoop-Plattform. Weitere Informationen finden Sie unter [Apache Ranger](#). Dank der nativen Integration können Sie Ihren eigenen Apache Ranger verwenden, um eine detaillierte Datenzugriffskontrolle auf Amazon EMR durchzusetzen.

Dieser Abschnitt bietet eine konzeptionelle Übersicht über die Amazon-EMR-Integration in Apache Ranger. Außerdem werden die Voraussetzungen und Schritte zum Starten eines in Apache Ranger integrierten Amazon-EMR-Clusters beschrieben.

Die native Integration von Amazon EMR mit Apache Ranger bietet die folgenden Hauptvorteile:

- Präzise Zugriffskontrolle für Hive Metastore-Datenbanken und -Tabellen, mit der Sie Datenfilterungsrichtlinien auf Datenbank-, Tabellen- und Spaltenebene für Apache Spark- und Apache Hive-Anwendungen definieren können. Filterung und Datenmaskierung auf Zeilenebene werden von Hive-Anwendungen unterstützt.
- Die Möglichkeit, Ihre bestehenden Hive-Richtlinien direkt mit Amazon EMR for Hive-Anwendungen zu verwenden.
- Zugriffskontrolle auf Amazon S3-Daten auf Präfix- und Objektebene, sodass Sie Datenfilterrichtlinien für den Zugriff auf S3-Daten mithilfe des EMR-Dateisystems definieren können.
- Die Möglichkeit, CloudWatch Logs für zentralisierte Prüfungen zu verwenden.

- Amazon EMR installiert und verwaltet die Apache Ranger-Plugin Ihrem Namen.

Apache Ranger mit Amazon EMR

Apache Ranger ist ein Framework zur Aktivierung, Überwachung und Verwaltung einer umfassenden Datensicherheit auf der gesamten Hadoop-Plattform.

Apache Ranger bietet folgende Features:

- Zentralisierte Sicherheitsadministration zur Verwaltung aller sicherheitsrelevanten Aufgaben in einer zentralen Benutzeroberfläche oder mithilfe von REST. APIs
- Detaillierte Autorisierung zur Durchführung einer bestimmten Aktion oder Operation mit einer Hadoop-Komponente oder einem Hadoop-Tool, die über ein zentrales Administrationstool verwaltet wird.
- Eine standardisierte Autorisierungsmethode für alle Hadoop-Komponenten.
- Verbesserte Unterstützung für verschiedene Autorisierungsmethoden.
- Zentralisierte Prüfung des Benutzerzugriffs und der administrativen Aktionen (sicherheitsbezogen) innerhalb aller Komponenten von Hadoop.

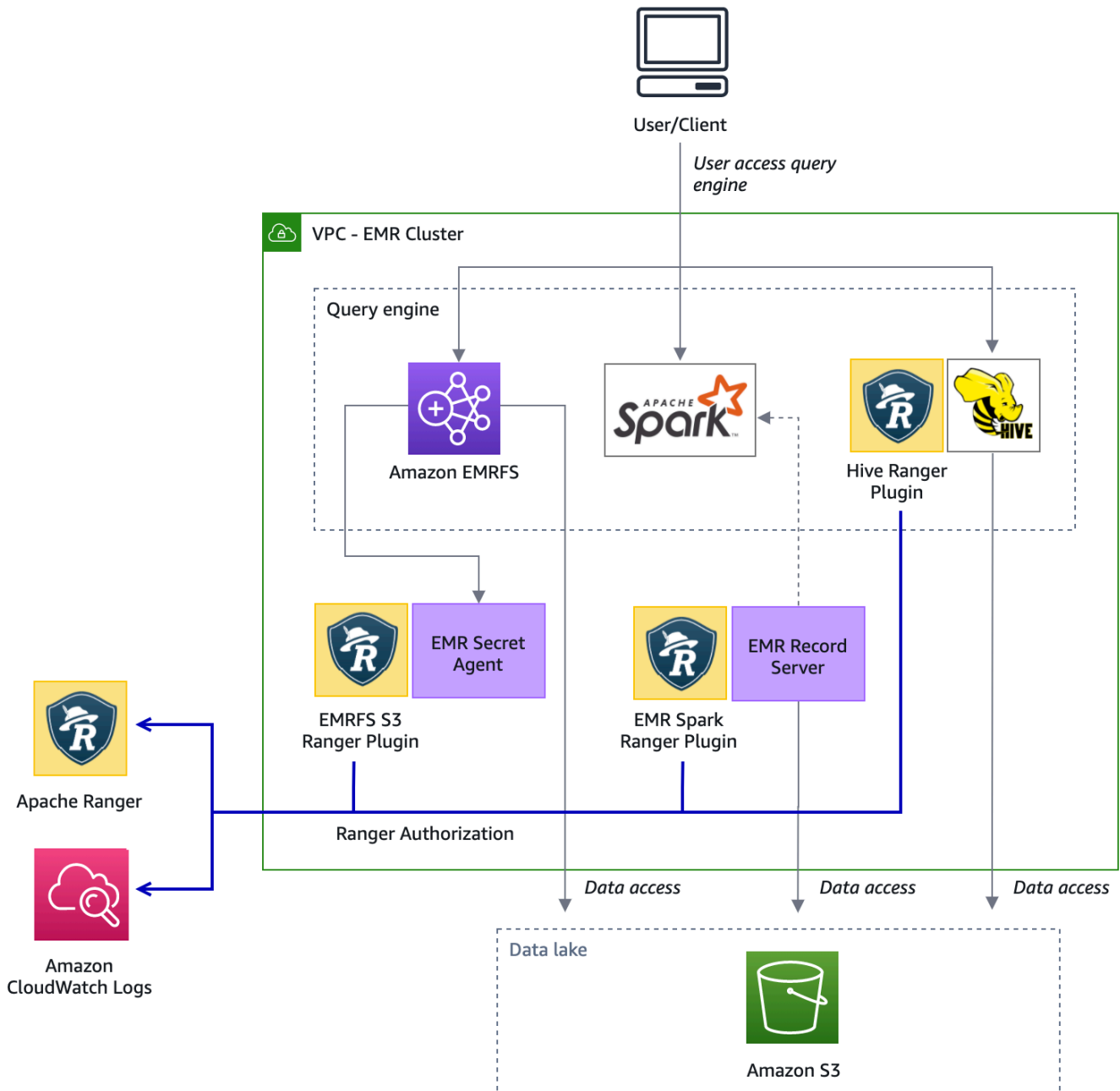
Apache Ranger verwendet zwei Schlüsselkomponenten für die Autorisierung:

- Apache-Ranger-Richtlinien-Admin-Server – Mit diesem Server können Sie die Autorisierungsrichtlinien für Hadoop-Anwendungen definieren. Bei der Integration mit Amazon EMR können Sie Richtlinien für Apache Spark und Hive für den Zugriff auf Hive Metastore und für den Zugriff auf Amazon-S3-Daten im [EMR File System \(EMRFS\)](#) definieren und durchsetzen. Sie können einen neuen Apache-Ranger-Richtlinien-Admin-Server einrichten oder einen vorhandenen Apache-Ranger-Richtlinien-Admin-Server für die Integration mit Amazon EMR verwenden.
- Apache-Ranger-Plugin – Dieses Plugin validiert den Zugriff eines Benutzers anhand der Autorisierungsrichtlinien, die im Apache-Ranger-Richtlinien-Admin-Server definiert sind. Amazon EMR installiert und konfiguriert das Apache-Ranger-Plugin automatisch für jede Hadoop-Anwendung, die in der Apache-Ranger-Konfiguration ausgewählt wurde.

Themen

- [Architektur der Amazon-EMR-Integration mit Apache Ranger](#)
- [Amazon EMR-Komponenten zur Verwendung mit Apache Ranger](#)

Architektur der Amazon-EMR-Integration mit Apache Ranger



Amazon EMR-Komponenten zur Verwendung mit Apache Ranger

Amazon EMR ermöglicht eine differenzierte Zugriffskontrolle mit Apache Ranger über die folgenden Komponenten. Im [Architekturdiagramm](#) finden Sie eine visuelle Darstellung dieser Amazon-EMR-Komponenten mit den Apache-Ranger-Plugins.

Secret-Agent – Der Secret-Agent speichert geheime Daten sicher und verteilt sie an andere Amazon-EMR-Komponenten oder Anwendungen. Bei diesen geheimen Daten („Secrets“) kann es sich beispielsweise um temporäre Anmeldeinformationen von Benutzern, um Verschlüsselungsschlüssel oder um Kerberos-Tickets handeln. Der Secret Agent läuft auf jedem Knoten im Cluster und fängt Aufrufe an den Instance Metadata Service ab. Für Anfragen an die Rollenmeldedaten des Instance-Profiles vergibt der Secret Agent je nach dem anfragenden Benutzer und den angeforderten Ressourcen Anmeldeinformationen, nachdem er die Anfrage mit dem EMRFS-S3-Ranger-Plugin autorisiert hat. Der Geheimagent wird als *emrsecretagent* Benutzer ausgeführt und schreibt Protokolle in das Verzeichnis `emr/secretagent/log`. Der Prozess benötigt eine bestimmte Zusammenstellung von iptables-Regeln, um zu funktionieren. Es ist wichtig sicherzustellen, dass iptables nicht deaktiviert ist. Wenn Sie die iptables-Konfiguration anpassen, müssen die NAT-Tabellenregeln beibehalten und unverändert bleiben.

EMR Datensatzserver – Der Datensatzserver empfängt Anforderungen für den Zugriff auf Daten von Spark. Anschließend autorisiert es Anfragen, indem es die angeforderten Ressourcen an das Spark-Ranger-Plugin für Amazon EMR weiterleitet. Der Datensatzserver liest Daten von Amazon S3 und gibt gefilterte Daten zurück, auf die der Benutzer gemäß der Ranger-Richtlinie zugreifen darf. Der Record-Server läuft auf jedem Knoten im Cluster als Benutzer `emr_record_server` und schreibt Protokolle in das Verzeichnis `-record-server. var/log/emr`

Überlegungen zur Verwendung von Amazon EMR mit Apache Ranger

Unterstützte Anwendungen für Amazon EMR mit Apache Ranger

Die Integration zwischen Amazon EMR und Apache Ranger, in der EMR Ranger-Plugins installiert, unterstützt derzeit die folgenden Anwendungen:

- Apache Spark (verfügbar mit EMR 5.32+ und EMR 6.3+)
- Apache Spark (verfügbar mit EMR 5.32+ und EMR 6.3+)
- S3-Zugriff über EMRFS (verfügbar mit EMR 5.32+ und EMR 6.3+)

Sie können auch die folgenden Anwendungen auf einem EMR-Cluster installieren und sie so konfigurieren, dass sie Ihren Sicherheitsanforderungen entsprechen:

- Apache Hadoop (verfügbar mit EMR 5.32+ und EMR 6.3+ einschließlich YARN und HDFS)
- Apache Livy (verfügbar mit EMR 5.32+ und EMR 6.3+)
- Apache Zeppelin (verfügbar mit EMR 5.32+ und EMR 6.3+)

- Apache Hue (verfügbar mit EMR 5.32+ und EMR 6.3+)
- Ganglia (verfügbar mit EMR 5.32+ und EMR 6.3+)
- HCatalog (Verfügbar mit EMR 5.32+ und EMR 6.3+)
- Mahout (verfügbar mit EMR 5.32+ und EMR 6.3+)
- MXNet (Verfügbar mit EMR 5.32+ und EMR 6.3+)
- TensorFlow (Verfügbar mit EMR 5.32+ und EMR 6.3+)
- Tez (Verfügbar mit EMR 5.32+ und EMR 6.3+)
- Trino (Verfügbar mit EMR 6.7+)
- ZooKeeper (Verfügbar mit EMR 5.32+ und EMR 6.3+)

Important

Die oben aufgeführten Anwendungen sind die einzigen Anwendungen, die derzeit unterstützt werden. Um die Clustersicherheit zu gewährleisten, dürfen Sie einen EMR-Cluster nur mit den Anwendungen in der obigen Liste erstellen, wenn Apache Ranger aktiviert ist.

Andere Anwendungen werden derzeit nicht unterstützt. Um die Sicherheit Ihres Clusters zu gewährleisten, führt der Versuch, andere Anwendungen zu installieren, zur Ablehnung Ihres Clusters.

AWS Glue Data Catalog und Open Table Formate wie Apache Hudi, Delta Lake und Apache Iceberg werden nicht unterstützt.

Unterstützte Amazon EMR-Funktionen mit Apache Ranger

Die folgenden Amazon EMR-Funktionen werden unterstützt, wenn Sie Amazon EMR mit Apache Ranger verwenden:

- Verschlüsselung bei Speicherung und Übertragung
- Kerberos-Authentifizierung (erforderlich)
- Instance-Gruppen, Instance-Flotten und Spot Instances
- Neukonfiguration von Anwendungen auf einem laufenden Cluster
- Serverseitige Verschlüsselung (SSE) von EMRFS

 Note

Die Amazon-EMR-Verschlüsselungseinstellungen regeln SSE. Weitere Informationen finden Sie unter [Verschlüsselungsoptionen](#).

Einschränkungen der Anwendung

Bei der Integration von Amazon EMR und Apache Ranger sind mehrere Einschränkungen zu beachten:

- Sie können die Konsole derzeit nicht verwenden, um eine Sicherheitskonfiguration zu erstellen, die die AWS Ranger-Integrationsoption in der spezifiziert. AWS GovCloud (US) Region Die Sicherheitskonfiguration kann mit der CLI durchgeführt werden.
- Kerberos muss in Ihrem Cluster installiert sein.
- Anwendungen UIs (Benutzeroberflächen) wie die YARN Resource Manager-Benutzeroberfläche, die NameNode HDFS-Benutzeroberfläche und die Livy-Benutzeroberfläche sind standardmäßig nicht mit Authentifizierung ausgestattet.
- Die HDFS-Standardberechtigungen umask sind so konfiguriert, dass erstellte Objekte standardmäßig auf `world wide readable` eingestellt sind.
- Amazon EMR unterstützt den Hochverfügbarkeitsmodus (mehrere Primärtypen) mit Apache Ranger nicht.
- Weitere Einschränkungen finden Sie unter Einschränkungen für jede Anwendung.

 Note

Die Amazon-EMR-Verschlüsselungseinstellungen regeln SSE. Weitere Informationen finden Sie unter [Verschlüsselungsoptionen](#).

Einschränkungen des Plugins

Jedes Plugin hat spezifische Einschränkungen. Die Einschränkungen des Apache-Hive-Plugins finden Sie unter [Einschränkungen des Apache-Hive-Plugins](#). Die Einschränkungen des Apache-Spark-Plugins finden Sie unter [Einschränkungen des Apache-Spark-Plugins](#). Die Einschränkungen des EMRFS-S3-Plugins finden Sie unter [Einschränkungen des EMRFS-S3-Plugins](#).

Amazon EMR für Apache Ranger einrichten

Bevor Sie Apache Ranger installieren, überprüfen Sie die Informationen in diesem Abschnitt, um sicherzustellen, dass Amazon EMR ordnungsgemäß konfiguriert ist.

Themen

- [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#)
- [IAM-Rollen für die native Integration mit Apache Ranger](#)
- [Erstellen einer EMR-Sicherheitskonfiguration](#)
- [Speichern Sie TLS-Zertifikate in AWS Secrets Manager](#)
- [Starten Sie einen EMR-Cluster mit Apache Ranger](#)
- [Zeppelin für Apache-Ranger-fähige Amazon-EMR-Cluster konfigurieren](#)
- [Bekannte Probleme bei der Amazon EMR-Integration](#)

Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein

Für die Amazon-EMR-Integration müssen die Apache-Ranger-Anwendungs-Plugins über TLS/SSL mit dem Admin-Server kommunizieren.

Voraussetzung: SSL-Aktivierung des Ranger-Admin-Servers

Apache Ranger auf Amazon EMR erfordert bidirektionale SSL-Kommunikation zwischen Plugins und dem Ranger-Admin-Server. Um sicherzustellen, dass Plugins über SSL mit dem Apache Ranger-Server kommunizieren, aktivieren Sie das folgende Attribut in ranger-admin-site.xml auf dem Ranger Admin-Server.

```
<property>
  <name>ranger.service.https.attrib.ssl.enabled</name>
  <value>true</value>
</property>
```

Darüber hinaus sind die folgenden Konfigurationen erforderlich.

```
<property>
  <name>ranger.https.attrib.keystore.file</name>
  <value>_<PATH_TO_KEYSTORE>_</value>
```

```
</property>

<property>
  <name>ranger.service.https.attrib.keystore.file</name>
  <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
  <name>ranger.service.https.attrib.keystore.pass</name>
  <value>_<KEYSTORE_PASSWORD>_</value>
</property>

<property>
  <name>ranger.service.https.attrib.keystore.keyalias</name>
  <value><PRIVATE_CERTIFICATE_KEY_ALIAS></value>
</property>

<property>
  <name>ranger.service.https.attrib.clientAuth</name>
  <value>want</value>
</property>

<property>
  <name>ranger.service.https.port</name>
  <value>6182</value>
</property>
```

TLS-Zertifikate für die Apache Ranger-Integration mit Amazon EMR

Die Apache-Ranger-Integration mit Amazon EMR erfordert, dass der Datenverkehr von Amazon-EMR-Knoten zum Ranger-Admin-Server mit TLS verschlüsselt wird und dass Ranger-Plugins sich beim Apache-Ranger-Server mithilfe der wechselseitigen TLS-Authentifizierung authentifizieren. Der Amazon-EMR-Service benötigt das öffentliche Zertifikat Ihres Ranger-Admin-Servers (wie im vorherigen Beispiel angegeben) und das private Zertifikat.

Zertifikate für das Apache-Ranger-Plugin

Öffentliche TLS-Zertifikate für das Apache Ranger-Plugin müssen für den Apache-Ranger-Admin-Server zugänglich sein, um zu überprüfen, wann die Plugins eine Verbindung herstellen. Es gibt drei verschiedene Methoden, dies zu tun.

Methode 1: Konfigurieren Sie einen Truststore auf dem Apache-Ranger-Admin-Server

Füllen Sie die folgenden Konfigurationen in ranger-admin-site .xml aus, um einen Truststore zu konfigurieren.

```
<property>
  <name>ranger.truststore.file</name>
  <value><LOCATION TO TRUSTSTORE></value>
</property>

<property>
  <name>ranger.truststore.password</name>
  <value><PASSWORD FOR TRUSTSTORE></value>
</property>
```

Methode 2: Laden Sie das Zertifikat in den Truststore von Java cacerts

Wenn Ihr Ranger- Admin-Server in seinen JVM-Optionen keinen Truststore angibt, können Sie die öffentlichen Plugin-Zertifikate im Standard-Cacerts-Speicher ablegen.

Methode 3: Erstellen Sie einen Truststore und geben Sie ihn als Teil der JVM-Optionen an

Ändern Sie die {RANGER_HOME_DIRECTORY}/ews/ranger-admin-services.sh innerhalb von JAVA_OPTS, dass sie "-Djavax.net.ssl.trustStore=<TRUSTSTORE_LOCATION>" und "-Djavax.net.ssl.trustStorePassword=<TRUSTSTORE_PASSWORD>" einschließt. Fügen Sie beispielsweise die folgende Zeile nach dem vorhandenen JAVA_OPTS hinzu.

```
JAVA_OPTS=" ${JAVA_OPTS} -Djavax.net.ssl.trustStore=${RANGER_HOME}/truststore/
truststore.jck -Djavax.net.ssl.trustStorePassword=changeit"
```

Note

Diese Spezifikation kann das Truststore-Passwort offenlegen, wenn sich ein Benutzer beim Apache- Ranger Admin-Server anmelden und laufende Prozesse sehen kann, z. B. wenn er den ps-Befehl verwendet.

Verwenden selbstsignierter Zertifikate

Selbstsignierte Zertifikate werden als Zertifikate nicht empfohlen. Selbstsignierte Zertifikate können nicht gesperrt werden, und selbstsignierte Zertifikate entsprechen möglicherweise nicht den internen Sicherheitsanforderungen.

Installation der Servicedefinition für die Ranger-Integration mit Amazon EMR

Eine Servicedefinition wird vom Ranger-Admin-Server verwendet, um die Attribute von Richtlinien für eine Anwendung zu beschreiben. Die Richtlinien werden dann in einem Richtlinien-Repository gespeichert, sodass die Clients sie herunterladen können.

Um Servicedefinitionen konfigurieren zu können, müssen REST-Aufrufe an den Ranger-Admin-Server getätigt werden. Informationen zu den [Anforderungen finden Sie im folgenden Abschnitt unter Apache Ranger Public APIs v2](#). APIs

Installation der Servicedefinition von Apache Spark

Informationen zur Installation der Servicedefinition von Apache Spark finden Sie unter [Apache Spark-Plugin für die Ranger-Integration mit Amazon EMR](#).

Installation der EMRFS-Servicedefinition

Informationen zur Installation der S3-Servicedefinition für Amazon EMR finden Sie unter [EMRFS S3-Plugin für die Ranger-Integration mit Amazon EMR](#).

Verwenden der Hive-Servicedefinition

Apache Hive kann die bestehende Ranger-Servicedefinition verwenden, die im Lieferumfang von Apache Ranger 2.0 und höher enthalten ist. Weitere Informationen finden Sie unter [Apache Hive-Plugin für die Ranger-Integration mit Amazon EMR](#).

Regeln für den Netzwerkverkehr für die Integration mit Amazon EMR

Wenn Apache Ranger in Ihren EMR-Cluster integriert ist, muss der Cluster mit zusätzlichen Servern und AWS kommunizieren.

Alle Amazon-EMR-Knoten, einschließlich Core- und Aufgabenknoten, müssen in der Lage sein, mit den Apache-Ranger-Admin-Servern zu kommunizieren, um Richtlinien herunterzuladen. Wenn Ihr Apache Ranger Admin auf Amazon läuft EC2, müssen Sie die Sicherheitsgruppe aktualisieren, um Datenverkehr vom EMR-Cluster entgegennehmen zu können.

Zusätzlich zur Kommunikation mit dem Ranger Admin-Server müssen alle Knoten in der Lage sein, mit den folgenden Diensten zu kommunizieren: AWS

- Amazon S3
- AWS KMS (bei Verwendung von EMRFS SSE-KMS)
- Amazon CloudWatch

- AWS STS

Wenn Sie planen, Ihren EMR-Cluster in einem privaten Subnetz auszuführen, konfigurieren Sie die VPC so, dass sie mit diesen Services entweder über [AWS PrivateLink und VPC-Endpunkte](#) im Amazon-VPC-Benutzerhandbuch oder mithilfe der [Network Address Translation \(NAT\)-Instance](#) im Amazon-VPC-Benutzerhandbuch kommunizieren kann.

IAM-Rollen für die native Integration mit Apache Ranger

Die Integration zwischen Amazon EMR und Apache Ranger basiert auf drei Schlüsselrollen, die Sie erstellen sollten, bevor Sie Ihren Cluster starten:

- Ein benutzerdefiniertes EC2 Amazon-Instance-Profil für Amazon EMR
- Eine IAM-Rolle für Apache Ranger Engines
- Eine IAM-Rolle für andere Dienste AWS

Dieser Abschnitt bietet eine Übersicht über diese Rollen und die Richtlinien, die Sie für jede dieser IAM-Rollen einschließen müssen. Informationen zum Erstellen dieser Rollen finden Sie unter [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#).

EC2 Instanzprofil für Amazon EMR

Amazon EMR verwendet eine IAM-Servicerolle, um in Ihrem Namen Aktionen zur Bereitstellung und Verwaltung von Clustern durchzuführen. Die Servicerolle für EC2 Cluster-Instances, auch EC2 Instance-Profil für Amazon EMR genannt, ist eine spezielle Art von Servicerolle, die jeder EC2 Instance in einem Cluster beim Start zugewiesen wird.

Um Berechtigungen für die EMR-Cluster-Interaktion mit Amazon S3 S3-Daten und mit dem durch Apache Ranger und andere AWS Dienste geschützten Hive-Metastore zu definieren, definieren Sie ein benutzerdefiniertes EC2 Instance-Profil, das anstelle des `EMR_EC2_DefaultRole` beim Starten Ihres Clusters verwendet werden soll.

Weitere Informationen erhalten Sie unter [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#) und [Passen Sie IAM-Rollen mit Amazon EMR an](#).

Sie müssen die folgenden Anweisungen zum EC2 Standard-Instance-Profil für Amazon EMR hinzufügen, um Sitzungen taggen und auf die zugreifen zu können, in AWS Secrets Manager denen TLS-Zertifikate gespeichert sind.

```

{
  "Sid": "AllowAssumeOfRolesAndTagging",
  "Effect": "Allow",
  "Action": ["sts:TagSession", "sts:AssumeRole"],
  "Resource": [
    "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_ENGINE-  

    PLUGIN_DATA_ACCESS_ROLE_NAME>",
    "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_USER_ACCESS_ROLE_NAME>"
  ],
},
{
  "Sid": "AllowSecretsRetrieval",
  "Effect": "Allow",
  "Action": "secretsmanager:GetSecretValue",
  "Resource": [
    "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<PLUGIN_TLS_SECRET_NAME>*",
    "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<ADMIN_RANGER_SERVER_TLS_SECRET_NAME>"
  ]
}

```

Note

Vergessen Sie bei den Secrets- Manager-Berechtigungen nicht den Platzhalter („*“) am Ende des geheimen Namens, da Ihre Anfragen sonst fehlschlagen. Der Platzhalter gilt für geheime Versionen.

Note

Beschränken Sie den Geltungsbereich der AWS Secrets Manager Richtlinie auf die Zertifikate, die für die Bereitstellung erforderlich sind.

IAM-Rolle für Apache Ranger

Diese Rolle stellt Anmeldeinformationen für vertrauenswürdige Ausführungs-Engines wie Apache Hive und Amazon EMR Record Server bereit, um auf Amazon-S3-Daten zuzugreifen. Verwenden Sie

nur diese Rolle, um auf Amazon-S3-Daten, einschließlich aller KMS-Schlüssel, zuzugreifen, wenn Sie S3 SSE-KMS verwenden.

Diese Rolle muss mit der im folgenden Beispiel angegebenen Mindestrichtlinie erstellt werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CloudwatchLogsPermissions",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:logs:<REGION>:<AWS_ACCOUNT_ID>:<CLOUDWATCH_LOG_GROUP_NAME_IN_SECURITY_CONFIGURATION>:"
      ]
    },
    {
      "Sid": "BucketPermissionsInS3Buckets",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
        "arn:aws:s3:::amzn-s3-demo-bucket2"
      ]
    },
    {
      "Sid": "ObjectPermissionsInS3Objects",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
```

```

    * "arn:aws:s3:::amzn-s3-demo-bucket1/*",
      "arn:aws:s3:::amzn-s3-demo-bucket2/*"
  * ]
}
]
}

```

Important

Das Sternchen „*“ am Ende der CloudWatch Protokollressource muss enthalten sein, um Schreibberechtigungen für die Protokolldatenströme zu erteilen.

Note

Wenn Sie die EMRFS-Konsistenzansicht oder die S3-SSE-Verschlüsselung verwenden, fügen Sie den DynamoDB-Tabellen und KMS-Schlüsseln Berechtigungen hinzu, damit die Ausführungsmodule mit diesen Engines interagieren können.

Die IAM-Rolle für Apache Ranger wird von der EC2 Instanzprofilrolle übernommen. Verwenden Sie das folgende Beispiel, um eine Vertrauensrichtlinie zu erstellen, die es ermöglicht, dass die IAM-Rolle für Apache Ranger von der EC2 Instance-Profilrolle übernommen wird.

```

{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2_INSTANCE_PROFILE_ROLE_NAME eg.
EMR_EC2_DefaultRole>"
  },
  "Action": ["sts:AssumeRole", "sts:TagSession"]
}

```

IAM-Rolle für andere AWS Services für die Amazon EMR-Integration

Diese Rolle bietet Benutzern, denen Execution Engines nicht vertraut werden, bei Bedarf Anmeldeinformationen für die Interaktion mit AWS Diensten. Verwenden Sie diese IAM-Rolle nicht, um Zugriff auf Amazon-S3-Daten zu gewähren, es sei denn, es handelt sich um Daten, auf die alle Benutzer zugreifen können sollten.

Diese Rolle wird von der EC2 Instanzprofilrolle übernommen. Verwenden Sie das folgende Beispiel, um eine Vertrauensrichtlinie zu erstellen, die es ermöglicht, dass die IAM-Rolle für Apache Ranger von der EC2 Instance-Profilrolle übernommen wird.

```
{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2_INSTANCE_PROFILE_ROLE_NAME eg.
EMR_EC2_DefaultRole>"
  },
  "Action": ["sts:AssumeRole", "sts:TagSession"]
}
```

Überprüfen Sie Ihre Berechtigungen für die Amazon EMR-Integration mit Apache Ranger

Anweisungen zum Überprüfen von Berechtigungen finden Sie unter [Fehlerbehebung für Apache Ranger](#).

Erstellen einer EMR-Sicherheitskonfiguration

Eine Amazon-EMR-Sicherheitskonfiguration für Apache Ranger erstellen

Bevor Sie einen in Apache Ranger integrierten Amazon-EMR-Cluster starten, erstellen Sie eine Sicherheitskonfiguration.

Console

So erstellen Sie eine Sicherheitskonfiguration, die die AWS -Ranger-Integrationsoption angibt

1. Wählen Sie in der Amazon-EMR-Konsole die Optionen Sicherheitskonfigurationen und dann Erstellen aus.
2. Geben Sie in Name (Name) einen Namen für die Sicherheitskonfiguration ein. Verwenden Sie diesen Namen zum Angeben der Sicherheitskonfiguration, wenn Sie einen Cluster erstellen.
3. Wählen Sie unter AWS -Ranger-Integration die Option Aktivieren einer von Apache Ranger verwalteten feinkörnigen Zugriffskontrolle.
4. Wählen Sie Ihre IAM-Rolle für die Apache Ranger, die angewendet werden soll. Weitere Informationen finden Sie unter [IAM-Rollen für die native Integration mit Apache Ranger](#).
5. Wählen Sie eine IAM-Rolle für andere AWS -Services aus, die angewendet werden soll.

6. Konfigurieren Sie die Plugins so, dass sie eine Verbindung zum Ranger Admin-Server herstellen, indem Sie den Secrets Manager Manager-ARN für den Admin-Server und die Adresse eingeben.
7. Wählen Sie die Anwendungen aus, um Ranger-Plugins zu konfigurieren. Geben Sie den Secrets Manager ARN ein, der das private TLS-Zertifikat für das Plugin enthält.

Wenn Sie Apache Spark oder Apache Hive nicht konfigurieren und diese als Anwendung für Ihren Cluster ausgewählt wurden, schlägt die Anfrage fehl.

8. Richten Sie weitere Sicherheitskonfigurationsoptionen ein wie erforderlich. Wählen Sie Create (Erstellen) aus. Sie müssen die Kerberos-Authentifizierung mit dem Cluster-spezifischen oder externen KDC aktivieren.

Note

Sie können die Konsole derzeit nicht verwenden, um eine Sicherheitskonfiguration zu erstellen, die die AWS Ranger-Integrationsoption in der AWS GovCloud (US) Region spezifiziert. Die Sicherheitskonfiguration kann mit der CLI durchgeführt werden.

CLI

Wie Sie eine Sicherheitskonfiguration für die Apache Ranger-Integration erstellen

1. Ersetzen Sie es **<ACCOUNT ID>** durch Ihre AWS Konto-ID.
2. Ersetzen Sie **<REGION>** durch die Region, in der sich die Ressource befindet.
3. Geben Sie einen Wert für `TicketLifetimeInHours` an, um den Zeitraum zu anzugeben, für den ein vom KDC ausgestelltes Kerberos-Ticket gültig ist.
4. Geben Sie die Adresse des Ranger-Admin-Servers für `AdminServerURL` an.

```
{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24
      }
    }
  }
}
```

```

},
"AuthorizationConfiguration":{
  "RangerConfiguration":{
    "AdminServerURL":"https://_<RANGER ADMIN SERVER IP>_:6182",
    "RoleForRangerPluginsARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<RANGER PLUGIN
DATA ACCESS ROLE NAME>_",
    "RoleForOtherAWSServicesARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<USER
ACCESS ROLE NAME>_",
    "AdminServerSecretARN":"arn:aws:secretsmanager:_<REGION>:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES ADMIN SERVERS PUBLIC TLS CERTIFICATE
WITHOUT VERSION>_",
    "RangerPluginConfigurations":[
      {
        "App":"Spark",
        "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES SPARK PLUGIN PRIVATE TLS CERTIFICATE
WITHOUT VERSION>_",
        "PolicyRepositoryName":"<SPARK SERVICE NAME eg. amazon-emr-spark>"
      },
      {
        "App":"Hive",
        "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES Hive PLUGIN PRIVATE TLS CERTIFICATE WITHOUT
VERSION>_",
        "PolicyRepositoryName":"<HIVE SERVICE NAME eg. Hivedev>"
      },
      {
        "App":"EMRFS-S3",
        "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES EMRFS S3 PLUGIN PRIVATE TLS CERTIFICATE
WITHOUT VERSION>_",
        "PolicyRepositoryName":"<EMRFS S3 SERVICE NAME eg amazon-emr-emrfs>"
      },
      {
        "App":"Trino",
        "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>:_<ACCOUNT
ID>_:secret:_<SECRET NAME THAT PROVIDES TRINO PLUGIN PRIVATE TLS CERTIFICATE
WITHOUT VERSION>_",
        "PolicyRepositoryName":"<TRINO SERVICE NAME eg amazon-emr-trino>"
      }
    ],
    "AuditConfiguration":{
      "Destinations":{
        "AmazonCloudWatchLogs":{

```

```

        "CloudWatchLogGroup": "arn:aws:logs:<REGION>:_<ACCOUNT ID>_:log-
group:_<LOG GROUP NAME FOR AUDIT EVENTS>_"
    }
}
}
}
}
}
}
}
}
}

```

Dies PolicyRespositoryNames sind die Dienstnamen, die in Ihrem Apache Ranger Admin angegeben sind.

Erstellen Sie eine Amazon-EMR-Datei für die Sicherheitskonfiguration mit dem folgenden Befehl. Ersetzen Sie Sicherheitskonfiguration durch einen beliebigen Namen. Wählen Sie diese Konfiguration beim Erstellen Ihres Clusters nach Namen aus.

```

aws emr create-security-configuration \
--security-configuration file:///./security-configuration.json \
--name security-configuration

```

Konfigurieren von zusätzlichen Sicherheits-Features

Um sicherzustellen, dass Amazon EMR sicher in Apache Ranger integriert ist, konfigurieren Sie die folgenden EMR-Sicherheits-Features:

- Aktivieren Sie die Kerberos-Authentifizierung über den Cluster-spezifischen oder externen KDC. Detaillierte Anweisungen finden Sie unter [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#).
- (Optional) Aktivieren Sie die Verschlüsselung von Daten während der Übertragung oder im Ruhezustand. Weitere Informationen finden Sie unter [Verschlüsselungsoptionen für Amazon EMR](#).

Weitere Informationen finden Sie unter [Sicherheit in Amazon EMR](#).

Speichern Sie TLS-Zertifikate in AWS Secrets Manager

Die auf einem Amazon-EMR-Cluster installierten Ranger-Plugins und der Ranger- Admin-Server müssen über TLS kommunizieren, um sicherzustellen, dass Richtliniendaten und andere gesendete Informationen nicht gelesen werden können, wenn sie abgefangen werden. EMR schreibt außerdem vor, dass sich die Plugins beim Ranger-Admin-Server authentifizieren, indem sie ein eigenes TLS-

Zertifikat bereitstellen und eine bidirektionale TLS-Authentifizierung durchführen. Für dieses Setup mussten vier Zertifikate erstellt werden: zwei Paare von privaten und öffentlichen TLS-Zertifikaten. Anweisungen zur Installation des Zertifikats auf Ihrem Ranger Admin-Server finden Sie unter [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#). Um die Einrichtung abzuschließen, benötigen die auf dem EMR-Cluster installierten Ranger-Plugins zwei Zertifikate: das öffentliche TLS-Zertifikat Ihres Admin-Servers und das private Zertifikat, das das Plugin zur Authentifizierung gegenüber dem Ranger-Admin-Server verwendet. Um diese TLS-Zertifikate bereitzustellen, müssen sie sich in der EMR-Sicherheitskonfiguration befinden AWS Secrets Manager und in dieser bereitgestellt werden.

Note

Es wird dringend empfohlen, aber nicht vorgeschrieben, für jede Ihrer Anwendungen ein Zertifikatspaar zu erstellen, um die Auswirkungen zu begrenzen, falls eines der Plugin-Zertifikate kompromittiert wird.

Note

Sie müssen Zertifikate vor ihrem Ablaufdatum nachverfolgen und rotieren.

Zertifikatformat

Das Importieren der Zertifikate in das AWS Secrets Manager ist identisch, unabhängig davon, ob es sich um das private Plugin-Zertifikat oder das öffentliche Ranger-Administratorzertifikat handelt. Vor dem Import der TLS-Zertifikate müssen die Zertifikate im 509x PEM-Format vorliegen.

Ein Beispiel für ein öffentliches Zertifikat hat das folgende Format:

```
-----BEGIN CERTIFICATE-----  
...Certificate Body...  
-----END CERTIFICATE-----
```

Ein Beispiel für ein privates Zertifikat hat das folgende Format:

```
-----BEGIN PRIVATE KEY-----  
...Private Certificate Body...  
-----END PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
...Trust Certificate Body...
-----END CERTIFICATE-----
```

Das private Zertifikat sollte auch ein Vertrauenszertifikat enthalten.

Mit folgendem Befehl können Sie prüfen, ob die Zertifikate das richtige Format haben:

```
openssl x509 -in <PEM FILE> -text
```

Ein Zertifikat in AWS Secrets Manager importieren

Wenn Sie Ihr Geheimnis im Secrets Manager erstellen, wählen Sie Andere Art von Geheimnissen unter Geheimnistyp und fügen Sie Ihr PEM-codiertes Zertifikat in das Klartext-Feld ein.

The screenshot shows the AWS Secrets Manager console interface. On the left, a sidebar indicates the current step is 'Step 3: Configure rotation', with 'Step 4: Review' also visible. The main area is titled 'Select secret type' and contains four radio button options: 'Credentials for RDS database', 'Credentials for DocumentDB database', 'Credentials for Redshift cluster', and 'Other type of secrets (e.g. API key)'. The 'Other type of secrets' option is selected. Below this, the section 'Specify the key/value pairs to be stored in this secret' is shown with two tabs: 'Secret key/value' and 'Plaintext'. The 'Plaintext' tab is active, and a large text area contains a PEM-encoded certificate, starting with '-----BEGIN CERTIFICATE-----' and ending with '-----END CERTIFICATE-----'. The certificate body is a long string of base64-encoded text.

Starten Sie einen EMR-Cluster mit Apache Ranger

Bevor Sie einen Amazon-EMR-Cluster mit Apache Ranger starten, stellen Sie sicher, dass jede Komponente die folgenden Mindestanforderungen an die Version erfüllt:

- Amazon EMR 5.32.0 oder höher oder 6.3.0 oder höher. Es wird empfohlen, die neueste Amazon-EMR-Release-Version zu verwenden.
- Apache Ranger Admin-Server 2.x.

Führen Sie folgende Schritte aus.

- Installieren Sie Apache Ranger, wenn das noch nicht geschehen ist. Weitere Informationen finden Sie unter [Installation von Apache Ranger 0.5.0](#).
- Stellen Sie sicher, dass zwischen Ihrem Amazon-EMR-Cluster und dem Apache-Ranger-Admin-Server eine Netzwerkverbindung besteht. Siehe [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#)
- Erstellen Sie die erforderlichen IAM-Rollen. Siehe [IAM-Rollen für die native Integration mit Apache Ranger](#).
- Erstellen Sie eine EMR-Sicherheitskonfiguration für die Apache-Ranger-Installation. Weitere Informationen finden Sie unter [Erstellen einer EMR-Sicherheitskonfiguration](#).

Zeppelin für Apache-Ranger-fähige Amazon-EMR-Cluster konfigurieren

Das Thema behandelt die Konfiguration von [Apache Zeppelin](#) für einen Apache-Ranger-fähigen Amazon-EMR-Cluster, sodass Sie Zeppelin als Notizbuch für die interaktive Datenexploration verwenden können. Zeppelin ist in Amazon-EMR-Versionen 5.0.0 und höher enthalten. Frühere Versionen enthalten Zeppelin als Sandbox-Anwendung. Weitere Informationen finden Sie unter [Informationen zu Amazon-EMR-4.x-Versionen](#) in den Amazon-EMR-Versionshinweisen.

Standardmäßig ist Zeppelin mit einem Standard-Login und Passwort konfiguriert, was in einer Umgebung mit mehreren Mandanten nicht sicher ist.

Führen Sie für die Konfiguration von Zeppelin die folgenden Schritte aus.

1. Verändern Sie den Authentifizierungsmechanismus.

Ändern Sie die `shiro.ini`-Datei, um Ihren bevorzugten Authentifizierungsmechanismus zu implementieren. Zeppelin unterstützt Active Directory, LDAP, PAM und Knox SSO. Weitere Informationen finden Sie unter [Apache-Shiro-Authentifizierung für Apache Zeppelin](#).

2. Konfigurieren Sie Zeppelin so, dass es sich als Endbenutzer ausgibt

Wenn Sie Zeppelin erlauben, sich als Endbenutzer auszugeben, können von Zeppelin eingereichte Aufträge als dieser Endbenutzer ausgeführt werden. Fügen Sie die folgende Konfiguration zu `core-site.xml` hinzu:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "hadoop.proxyuser.zeppelin.hosts": "*",
      "hadoop.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

Fügen Sie als Nächstes die folgende Konfiguration zu `hadoop-kms-site.xml` in `/etc/hadoop/conf` hinzu:

```
[
  {
    "Classification": "hadoop-kms-site",
    "Properties": {
      "hadoop.kms.proxyuser.zeppelin.hosts": "*",
      "hadoop.kms.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

Sie können diese Konfigurationen auch mithilfe der Konsole zu Ihrem Amazon-EMR-Cluster hinzufügen, indem Sie die Schritte unter [Instance-Gruppe in der Konsole neu konfigurieren](#) befolgen.

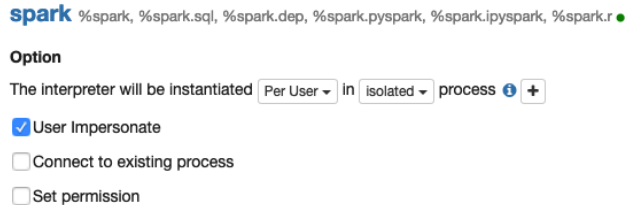
3. Erlauben Sie Zeppelin, als Endbenutzer `sudo` auszuführen

Erstellen Sie eine Datei `/etc/sudoers.d/90-zeppelin-user`, die folgendes enthält:

```
zeppelin ALL=(ALL) NOPASSWD:ALL
```


- Ändern Sie die Einstellungen der Interpreter, um Benutzeraufträge in ihren eigenen Prozessen auszuführen.

Konfigurieren Sie alle Interpreter so, dass sie die Interpreter „pro Benutzer“ in „isolierten“ Prozessen instanziiieren.



- Modifizieren Sie **zeppelin-env.sh**

Fügen Sie Folgendes zu `zeppelin-env.sh` hinzu, damit Zeppelin die Interpreter als Endbenutzer startet:

```
ZEPELIN_IMPERSONATE_USER=`echo ${ZEPELIN_IMPERSONATE_USER} | cut -d @ -f1`
export ZEPELIN_IMPERSONATE_CMD='sudo -H -u ${ZEPELIN_IMPERSONATE_USER} bash -c'
```

Fügen Sie Folgendes zu `zeppelin-env.sh` hinzu, um die standardmäßigen Notebookberechtigungen für den Ersteller auf Schreibgeschützt zu ändern:

```
export ZEPELIN_NOTEBOOK_PUBLIC="false"
```

Fügen Sie abschließend Folgendes hinzu, `zeppelin-env.sh` um den RecordServer EMR-Klassenpfad nach der ersten CLASSPATH Anweisung einzubeziehen:

```
export CLASSPATH="$CLASSPATH:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-connector-common.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-spark-connector.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-client.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-common.jar:/usr/share/aws/emr/record-server/lib/jars/secret-agent-interface.jar"
```

- Starten Sie Zeppelin neu.

Führen Sie für den folgenden Befehle aus, um Zeppelin neu zu starten:

```
sudo systemctl restart zeppelin
```

Bekannte Probleme bei der Amazon EMR-Integration

Bekannte Probleme

Es gibt ein bekanntes Problem in Amazon-EMR-Version 5.32, bei dem die Berechtigungen für `hive-site.xml` geändert wurden, sodass nur privilegierte Benutzer es lesen können, da möglicherweise Anmeldeinformationen darin gespeichert sind. Dies könnte Hue am Lesen von `hive-site.xml` hindern und dazu führen, dass Webseiten ständig neu geladen werden. Wenn dieses Problem auftritt, fügen Sie die folgende Konfiguration hinzu, um das Problem zu beheben:

```
[
  {
    "Classification": "hue-ini",
    "Properties": {},
    "Configurations": [
      {
        "Classification": "desktop",
        "Properties": {
          "server_group": "hive_site_reader"
        },
        "Configurations": [
        ]
      }
    ]
  }
]
```

Es gibt ein bekanntes Problem, dass das EMRFS-S3-Plugin für Apache Ranger das Security-Zone-Feature von Apache Ranger derzeit nicht unterstützt. Einschränkungen der Zugriffskontrolle, die mit dem Sicherheitszone-Feature definiert wurden, gelten nicht für Ihre Amazon-EMR-Cluster.

Anwendung Uls

Standardmäßig führen Benutzeroberflächen von Anwendungen keine Authentifizierung durch. Dazu gehören unter anderem die ResourceManager Benutzeroberfläche, die NodeManager Benutzeroberfläche und die Livy-Benutzeroberfläche. Darüber hinaus kann jeder Benutzer, der auf die Uls zugreifen kann, Informationen zu den Jobs aller anderen Benutzer einsehen.

Wenn dieses Verhalten nicht erwünscht ist, sollten Sie sicherstellen, dass eine Sicherheitsgruppe verwendet wird, um den Zugriff der Benutzer auf die Anwendung Uls einzuschränken.

HDFS-Standardberechtigungen

Standardmäßig erhalten die Objekte, die Benutzer in HDFS erstellen, weltweit lesbare Berechtigungen. Dies kann möglicherweise dazu führen, dass Daten von Benutzern gelesen werden, die keinen Zugriff darauf haben sollten. Gehen Sie wie folgt vor, um dieses Verhalten so zu ändern, dass die standardmäßigen Dateiberechtigungen nur vom Ersteller des Auftrags auf Lese- und Schreibzugriff festgelegt werden.

Geben Sie bei der Erstellung Ihres EMR-Clusters die folgende Konfiguration an:

```
[
  {
    "Classification": "hdfs-site",
    "Properties": {
      "dfs.namenode.acls.enabled": "true",
      "fs.permissions.umask-mode": "077",
      "dfs.permissions.superusergroup": "hdfsadmingroup"
    }
  }
]
```

Führen Sie außerdem die folgende Bootstrap-Aktion aus:

```
--bootstrap-actions Name='HDFS UMask Setup',Path=s3://elasticmapreduce/hdfs/umask/umask-main.sh
```

Apache Ranger-Plugins für Amazon EMR-Integrationsszenarien

Apache-Ranger-Plugin – Dieses Plugin validiert den Zugriff eines Benutzers anhand der Autorisierungsrichtlinien, die im Apache-Ranger-Richtlinien-Admin-Server definiert sind.

Themen

- [Apache Hive-Plugin für die Ranger-Integration mit Amazon EMR](#)
- [Apache Spark-Plugin für die Ranger-Integration mit Amazon EMR](#)
- [EMRFS S3-Plugin für die Ranger-Integration mit Amazon EMR](#)
- [Trino-Plugin für die Ranger-Integration mit Amazon EMR](#)

Apache Hive-Plugin für die Ranger-Integration mit Amazon EMR

Apache Hive ist eine beliebte Ausführungs-Engine innerhalb des Hadoop-Ökosystems. Amazon EMR bietet ein Apache-Ranger-Plugin, um detaillierte Zugriffskontrollen für Hive bereitstellen zu

können. Das Plugin ist mit Admin-Server-Version von Open-Source-Apache-Ranger 2.0 und höher kompatibel.

Themen

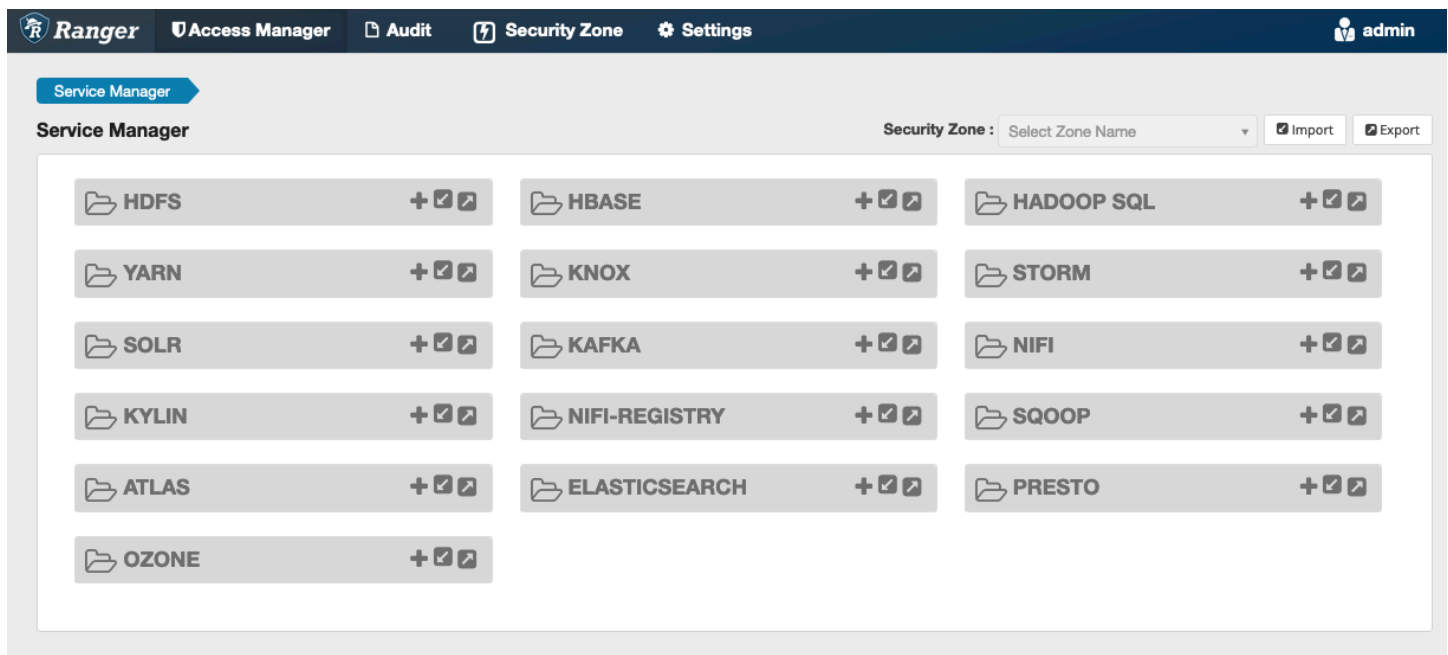
- [Unterstützte Features](#)
- [Installation der Servicekonfiguration](#)
- [Überlegungen](#)
- [Einschränkungen](#)

Unterstützte Features

Das Apache Ranger-Plugin für Hive on EMR unterstützt alle Funktionen des Open-Source-Plugins, einschließlich Zugriffskontrollen auf Datenbank-, Tabellen- und Spaltenebene sowie Zeilenfilterung und Datenmaskierung. Eine Tabelle mit Hive-Befehlen und den zugehörigen Ranger-Berechtigungen finden Sie unter Zuordnung von [Hive-Befehlen zu Ranger-Berechtigungen](#).

Installation der Servicekonfiguration

Das Apache Hive-Plug-in ist mit der vorhandenen Hive-Servicedefinition in Apache Hive Hadoop SQL kompatibel.



Wenn Sie keine Instance des Services unter Hadoop SQL haben, wie oben gezeigt, können Sie eine erstellen. Klicken Sie auf das + neben Hadoop SQL.

1. Servicename (falls angezeigt): Geben Sie den Servicennamen ein. Der vorgeschlagene Wert ist **amazonemrhive**. Notieren Sie sich diesen Servicennamen – er wird benötigt, wenn Sie eine EMR-Sicherheitskonfiguration erstellen.
2. Anzeigenname: Der Name, der für diesen Service angezeigt wird. Der vorgeschlagene Wert ist **amazonemrhive**.

The screenshot shows the Apache Ranger web interface. At the top is a navigation bar with 'Ranger', 'Access Manager', 'Audit', 'Security Zone', and 'Settings'. A user icon labeled 'admin' is in the top right. Below the navigation bar is a breadcrumb trail: 'Service Manager' > 'Create Service'. The main section is titled 'Create Service'. Under 'Service Details', there are five fields: 'Service Name *' (text input with 'amazonemrhive'), 'Display Name' (text input with 'amazonemrhive'), 'Description' (text area with 'Apache Hive policy repository for Amazon EMR'), 'Active Status' (radio buttons for 'Enabled' and 'Disabled', with 'Enabled' selected), and 'Select Tag Service' (a dropdown menu showing 'Select Tag Service').

Die Apache Hive Config Properties werden verwendet, um eine Verbindung zu Ihrem Apache Ranger Admin-Server mit einer HiveServer 2 herzustellen, um die auto Vervollständigung bei der Erstellung von Richtlinien zu implementieren. Die folgenden Eigenschaften müssen nicht korrekt sein, wenn Sie nicht über einen persistenten HiveServer 2-Prozess verfügen, und sie können mit beliebigen Informationen gefüllt werden.

- Benutzername: Geben Sie einen Benutzernamen für die JDBC-Verbindung zu einer Instanz einer HiveServer 2-Instanz ein.
- Passwort: Geben Sie das Passwort für den obigen Benutzernamen ein.
- jdbc.driver. ClassName: Geben Sie den Klassennamen der JDBC-Klasse für Apache Hive-Konnektivität ein. Sie können den Standardwert verwenden.
- jdbc.url: Geben Sie die JDBC-Verbindungszeichenfolge ein, die für die Verbindung mit 2 verwendet werden soll. HiveServer
- Allgemeiner Name für das Zertifikat: Das CN-Feld innerhalb des Zertifikats, das verwendet wird, um von einem Client-Plugin aus eine Verbindung zum Admin-Server herzustellen. Dieser Wert muss mit dem CN-Feld in Ihrem TLS-Zertifikat übereinstimmen, das für das Plugin erstellt wurde.

Config Properties :

Username *

Password *

jdbc.driverClassName *

jdbc.url *

Common Name for Certificate

Add New Configurations

Name	Value

Mit der Schaltfläche Verbindung testen wird getestet, ob die obigen Werte verwendet werden können, um erfolgreich eine Verbindung mit der 2-Instanz herzustellen. HiveServer Sobald der Service erfolgreich erstellt wurde, sollte der Service Manager wie folgt aussehen:

Ranger Access Manager Audit Security Zone Settings admin

Service Manager

Service Manager Security Zone :

HDFS + ⓘ ⌵	HBASE + ⓘ ⌵	HADOOP SQL + ⓘ ⌵ amazonemrhive ⓘ ⓘ ⌵
YARN + ⓘ ⌵	KNOX + ⓘ ⌵	STORM + ⓘ ⌵
SOLR + ⓘ ⌵	KAFKA + ⓘ ⌵	NIFI + ⓘ ⌵
KYLIN + ⓘ ⌵	NIFI-REGISTRY + ⓘ ⌵	SQOOP + ⓘ ⌵
ATLAS + ⓘ ⌵	ELASTICSEARCH + ⓘ ⌵	PRESTO + ⓘ ⌵
OZONE + ⓘ ⌵		

Überlegungen

Hive-Metadatenserver

Zum Schutz vor unbefugtem Zugriff können nur vertrauenswürdige Engines, insbesondere Hive und `emr_record_server`, auf den Hive-Metadatenserver zugreifen. Auf den Hive-Metadatenserver greifen auch alle Knoten im Cluster zu. Der erforderliche Port 9 083 ermöglicht allen Knoten den Zugriff auf den Hauptknoten.

Authentifizierung

Standardmäßig ist Apache Hive für die Authentifizierung mithilfe von Kerberos konfiguriert, wie in der EMR Security-Konfiguration konfiguriert. HiveServer2 kann so konfiguriert werden, dass Benutzer auch über LDAP authentifiziert werden. Weitere Informationen finden Sie unter [Implementieren der LDAP-Authentifizierung für Hive auf einem Amazon-EMR-Cluster mit mehreren Mandanten](#).

Einschränkungen

Die folgenden Einschränkungen gelten derzeit für das Apache Hive-Plugin auf Amazon EMR 5.x:

- Hive-Rollen werden derzeit nicht unterstützt. Die Anweisungen „Grant“ und „Revoke“ werden nicht unterstützt.
- Hive CLI wird nicht unterstützt. JDBC/Beeline ist der einzig autorisierte Weg, Hive zu verbinden.
- `hive.server2.builtin.udf.blacklist`Die Konfiguration sollte mit Daten gefüllt sein UDFs , die Sie für unsicher halten.

Apache Spark-Plugin für die Ranger-Integration mit Amazon EMR

Amazon EMR hat EMR integriert, RecordServer um eine differenzierte Zugriffskontrolle für SparkSQL zu ermöglichen. EMR RecordServer ist ein privilegierter Prozess, der auf allen Knoten eines Apache Ranger-fähigen Clusters ausgeführt wird. Wenn ein Spark-Treiber oder -Executor eine SparkSQL-Anweisung ausführt, durchlaufen alle Metadaten und Datenanforderungen die. RecordServer Weitere Informationen zu EMR RecordServer finden Sie [Amazon EMR-Komponenten zur Verwendung mit Apache Ranger](#) auf der Seite.

Themen

- [Unterstützte Features](#)
- [Stellen Sie die Servicedefinition erneut bereit, um INSERT-, ALTER- oder DDL-Anweisungen zu verwenden](#)

- [Installation der Servicedefinition](#)
- [Erstellen von SparkSQL-Richtlinien](#)
- [Überlegungen](#)
- [Einschränkungen](#)

Unterstützte Features

SQL-Aktion statement/Ranger	STATUS	Unterstützte Versionen für Amazon EMR
SELECT	Unterstützt	Ab 5.32
SHOW DATABASES	Unterstützt	Ab 5.32
SHOW_COLUMNS	Unterstützt	Ab 5.32
SHOW TABLES	Unterstützt	Ab 5.32
ANZEIGEN DER TABELLENEINGENSCHAFTEN	Unterstützt	Ab 5.32
DESCRIBE TABLE	Unterstützt	Ab 5.32
INSERT OVERWRITE	Unterstützt	Ab 5.34 und 6.4
INSERT INTO	Unterstützt	Ab 5.34 und 6.4
ALTER TABLE	Unterstützt	Ab 6.4
CREATE TABLE	Unterstützt	Ab 5.35 und 6.7
CREATE DATABASE	Unterstützt	Ab 5.35 und 6.7

SQL-Aktion statement/Ranger	STATUS	Unterstützte Versionen für Amazon EMR
DROP TABLE	Unterstützt	Ab 5.35 und 6.7
DROP DATABASE	Unterstützt	Ab 5.35 und 6.7
DROP VIEW	Unterstützt	Ab 5.35 und 6.7
CREATE VIEW	Nicht unterstützt	

Die folgenden Feature werden bei der Verwendung von SparkSQL unterstützt:

- Eine detaillierte Zugriffskontrolle für Tabellen im Hive-Metastore und Richtlinien können auf Datenbank-, Tabellen- und Spaltenebene erstellt werden.
- Die Richtlinien von Apache Ranger können Richtlinien für die Gewährung und die Ablehnung von Benutzern und Gruppen beinhalten.
- Prüfereignisse werden an CloudWatch Logs übermittelt.

Stellen Sie die Servicedefinition erneut bereit, um INSERT-, ALTER- oder DDL-Anweisungen zu verwenden

Note

Ab Amazon EMR 6.4 können Sie Spark SQL mit den folgenden Anweisungen verwenden: INSERT INTO, INSERT OVERWRITE oder ALTER TABLE. Ab Amazon EMR 6.7 können Sie Spark SQL verwenden, um Datenbanken und Tabellen zu erstellen oder zu löschen. Wenn Sie bereits über eine Installation auf dem Apache Ranger-Server mit bereitgestellten Apache Spark-Servicedefinitionen verfügen, verwenden Sie den folgenden Code, um die Servicedefinitionen erneut bereitzustellen.

```
# Get existing Spark service definition id calling Ranger REST API and JSON
processor
curl --silent -f -u <admin_user_login>:<password_for_ranger_admin_user> \
-H "Accept: application/json" \
```

```
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>:6182/service/public/v2/api/servicedef/
name/amazon-emr-spark' | jq .id

# Download the latest Service definition
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-spark.json

# Update the service definition using the Ranger REST API
curl -u <admin_user_login>:<password_for_ranger_admin_user> -X PUT -d @ranger-
servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>:6182/service/public/v2/api/
servicedef/<Spark service definition id from step 1>'
```

Installation der Servicedefinition

Die Installation der Trino-Servicedefinition erfordert die Einrichtung des Ranger-Admin-Servers. Siehe [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#).

Gehen Sie wie folgt vor, um die Apache-Spark-Servicedefinition zu installieren:

Schritt 1: SSH-Verbindung zum Apache-Ranger-Admin-Server herstellen

Zum Beispiel:

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

Schritt 2: Die Servicedefinition und das Apache-Ranger-Admin-Server-Plugin herunterladen

Laden Sie die Servicedefinition in einem temporären Verzeichnis herunter. Diese Servicedefinition wird von Ranger-2.x-Versionen unterstützt.

```
mkdir /tmp/emr-spark-plugin/
cd /tmp/emr-spark-plugin/

wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-spark-plugin-2.x.jar
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-spark.json
```

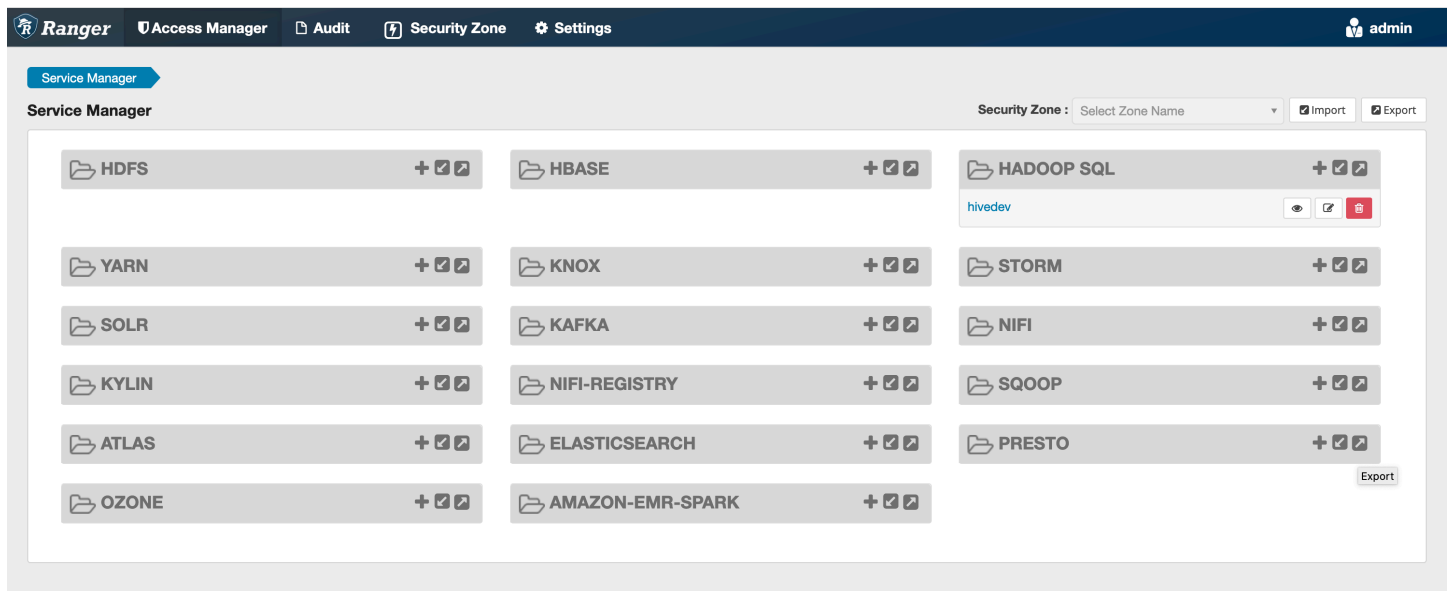
Schritt 3: Das Apache-Spark-Plugin für Amazon EMR installieren

```
export RANGER_HOME=.. # Replace this Ranger Admin's home directory eg /usr/lib/ranger/
ranger-2.0.0-admin
mkdir $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/amazon-emr-spark
mv ranger-spark-plugin-2.x.jar $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/
amazon-emr-spark
```

Schritt 4: Die Apache-Spark-Servicedefinition für Amazon EMR registrieren

```
curl -u *(<admin users login>*:*_**_password_**_for_**_ranger admin user_**_)_* -X
  POST -d @ranger-servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*(<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

Wenn dieser Befehl erfolgreich ausgeführt wird, sehen Sie in der Ranger-Admin-Benutzeroberfläche einen neuen Servicenamens „AMAZON-EMR-SPARK“, wie im folgenden Image gezeigt (Ranger-Version 2.0 wird angezeigt).



Schritt 5: Erstellen Sie eine Instanz der AMAZON-EMR-SPARK Anwendung

ServiceName (falls angezeigt): Der Servicename, der verwendet wird. Der vorgeschlagene Wert ist **amazonemrspark**. Notieren Sie sich diesen Servicennamen, da er für die Erstellung einer EMR-Sicherheitskonfiguration benötigt wird.

Anzeigenname: Der Name, der für diese Instance angezeigt werden soll. Der vorgeschlagene Wert ist **amazonemrspark**.

Allgemeiner Name für das Zertifikat: Das CN-Feld innerhalb des Zertifikats, das verwendet wird, um von einem Client-Plugin aus eine Verbindung zum Admin-Server herzustellen. Dieser Wert muss mit dem CN-Feld in Ihrem TLS-Zertifikat übereinstimmen, das für das Plugin erstellt wurde.

Service Manager > Create Service

Create Service

Service Details :

Service Name * amazonemrspark

Display Name amazonemrspark

Description

Active Status ☒ Enabled ☐ Disabled

Select Tag Service Select Tag Service

Config Properties :

Common Name for Certificate CNofCertificate

Add New Configurations

Name	Value

+

Test Connection

Add Cancel

Note

Das TLS-Zertifikat für dieses Plugin sollte im Trust-Store auf dem Ranger-Admin-Server registriert worden sein. Weitere Details finden Sie unter [TLS-Zertifikate für die Apache Ranger-Integration mit Amazon EMR](#).

Erstellen von SparkSQL-Richtlinien

Beim Erstellen einer neuen Richtlinie müssen folgende Felder ausgefüllt werden:

Richtlinienname: Der Name dieser Richtlinie.

Richtlinienbezeichnung: Eine Bezeichnung, die Sie dieser Richtlinie hinzufügen können.

Datenbank: Die Datenbank, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Tabellen.

Tabelle: Die Tabellen, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Tabellen.

EMR Spark-Spalte: Die Spalten, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Spalten.

Beschreibung: Eine Beschreibung dieser Richtlinie.

The screenshot shows the 'Create Policy' page in the Apache Ranger console. The breadcrumb trail is 'Service Manager > amazonemrspark Policies > Create Policy'. The page title is 'Create Policy'. Under 'Policy Details', the 'Policy Type' is set to 'Access'. There is a button 'Add Validity Period'. The 'Policy Name' field contains 'PolicyName' with an 'enabled' toggle. The 'Policy Label' field contains 'Policy Label'. There are three inclusion rules: 1) 'database' dropdown with 'default' selected and 'include' toggle; 2) 'table' dropdown with 'table' selected and 'include' toggle; 3) 'EMR Spark Column' field with '*' selected and 'include' toggle. The 'Description' field is empty. The 'Audit Logging' toggle is set to 'YES'.

Um die Benutzer und Gruppen anzugeben, geben Sie die Benutzer und Gruppen unten ein, um Berechtigungen zu erteilen. Sie können auch Ausnahmen für die Bedingungen Zulassen und Verweigern angeben.

Allow Conditions : hide ^

Select Role	Select Group	Select User		Delegate Admin	
Select Roles	× hadoop_analyst	× analyst1	add/edit permissions ☒ select ☐ × Add Permissions +	☐	×
+					

Exclude from Allow Conditions : hide ^

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	Select Users	Add Permissions +	☐	×
+					

Nachdem Sie die Bedingungen für das Zulassen und Verweigern angegeben haben, klicken Sie auf Speichern.

Überlegungen

Jeder Knoten im EMR-Cluster muss eine Verbindung zum Hauptknoten an Port 9 083 herstellen können.

Einschränkungen

Die folgenden Einschränkungen gelten derzeit für das Apache-Spark-Plugin:

- Der Record-Server stellt immer eine Verbindung zu HMS her, das auf einem Amazon-EMR-Cluster läuft. Konfigurieren Sie HMS für die Verbindung zum Remote-Modus, falls erforderlich. Sie sollten keine Konfigurationswerte in die Apache-Spark-Konfigurationsdatei Hive-site.xml einfügen.
- Tabellen, die mit Spark-Datenquellen auf CSV oder Avro erstellt wurden, können mit EMR nicht gelesen werden. RecordServer Verwenden Sie Hive, um Daten zu erstellen und zu schreiben, und lesen Sie sie mit Record.
- Delta Lake-, Hudi- und Iceberg-Tabellen werden nicht unterstützt.
- Benutzer müssen Zugriff auf die Standarddatenbank haben. Dies ist eine Voraussetzung für Apache Spark.
- Der Ranger-Admin-Server unterstützt die automatische Vervollständigung nicht.
- Das SparkSQL-Plugin für Amazon EMR unterstützt keine Zeilenfilter oder Datenmaskierung.

- Wenn Sie ALTER TABLE mit Spark SQL verwenden, muss ein Partitionsspeicherort das untergeordnete Verzeichnis eines Tabellenspeicherorts sein. Das Einfügen von Daten in eine Partition, deren Partitionsspeicherort sich von der Tabellenposition unterscheidet, wird nicht unterstützt.

EMRFS S3-Plugin für die Ranger-Integration mit Amazon EMR

Um die Bereitstellung von Zugriffskontrollen für Objekte in S3 auf einem Multi-Tenant-Cluster zu vereinfachen, bietet das EMRFS-S3-Plugin Zugriffskontrollen für die Daten in S3, wenn über EMRFS darauf zugegriffen wird. Sie können den Zugriff auf S3-Ressourcen auf Benutzer- und Gruppenebene zulassen.

Um dies zu erreichen, sendet EMRFS, wenn Ihre Anwendung versucht, auf Daten innerhalb von S3 zuzugreifen, eine Anfrage nach Anmeldeinformationen an den Secret-Agent-Prozess, wo die Anfrage anhand eines Apache-Ranger-Plugins authentifiziert und autorisiert wird. Wenn die Anfrage autorisiert ist, übernimmt der Secret Agent die IAM-Rolle für Apache-Ranger-Engines mit einer eingeschränkten Richtlinie, um Anmeldeinformationen zu generieren, die nur Zugriff auf die Ranger-Richtlinie haben, die den Zugriff gewährt hat. Die Anmeldeinformationen werden dann an EMRFS zurückgegeben, um auf S3 zuzugreifen.

Themen

- [Unterstützte Features](#)
- [Installation der Servicekonfiguration](#)
- [EMRFS-S3-Richtlinien erstellen](#)
- [Hinweise zur Verwendung von EMRFS-S3-Richtlinien](#)
- [Einschränkungen](#)

Unterstützte Features

Das EMRFS-S3-Plugin ermöglicht die Autorisierung auf Speicherebene. Richtlinien können erstellt werden, um Benutzern und Gruppen Zugriff auf S3-Buckets und -Präfixe zu gewähren. Die Autorisierung erfolgt nur für EMRFS.

Installation der Servicekonfiguration

Um die EMRFS-Servicedefinition zu installieren, müssen Sie den Ranger Admin-Server einrichten. Informationen zum Einrichten des Servers finden Sie unter [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#)

Gehen Sie wie folgt vor, um die Apache-Spark-Servicedefinition zu installieren.

Schritt 1: Stellen Sie eine SSH-Verbindung zum Apache-Ranger-Admin-Server her.

Zum Beispiel:

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

Schritt 2: Laden Sie die EMRFS-Servicedefinition herunter.

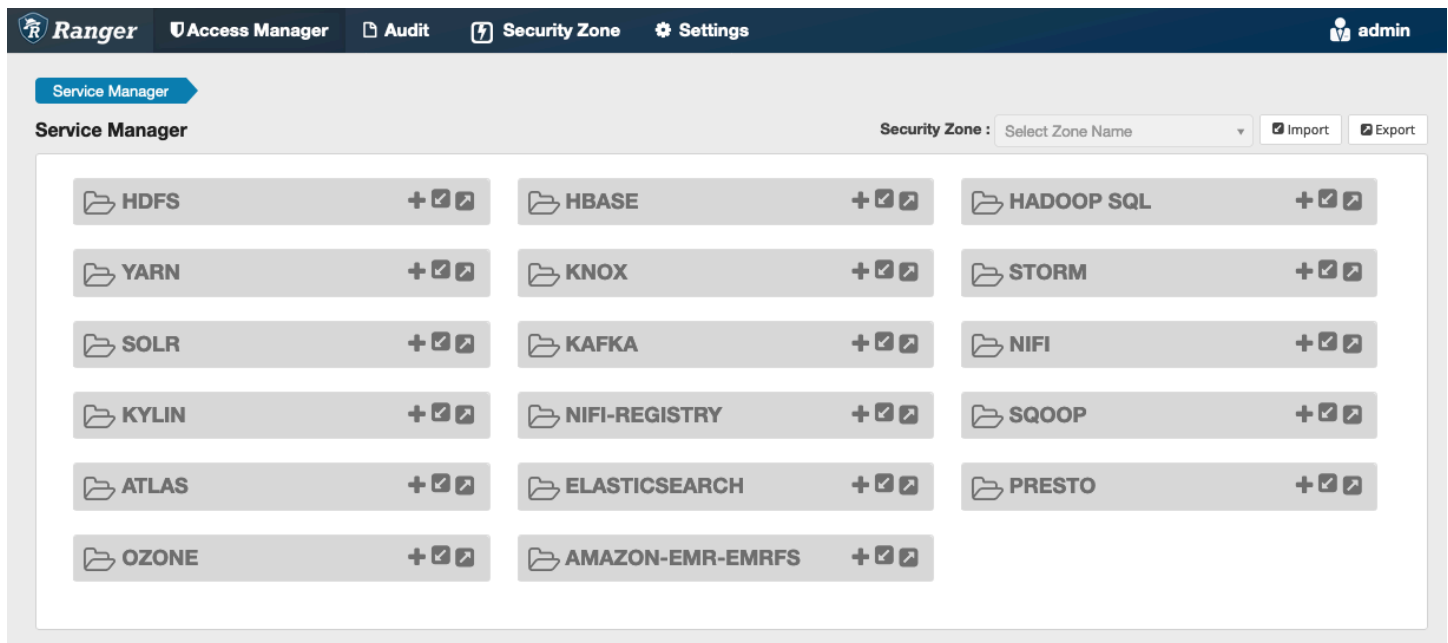
Laden Sie in einem temporären Verzeichnis die Amazon-EMR-Servicedefinition herunter. Diese Servicedefinition wird von Ranger-2.x-Versionen unterstützt.

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/ranger-servicedef-amazon-emr-emrfs.json
```

Schritt 3: Registrieren Sie die EMRFS S3-Servicedefinition.

```
curl -u *(<admin users login>*:*_**_password_ **_for_** _ranger admin user_**_)_* -X  
POST -d @ranger-servicedef-amazon-emr-emrfs.json \  
-H "Accept: application/json" \  
-H "Content-Type: application/json" \  
-k 'https://*(<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

Wenn dieser Befehl erfolgreich ausgeführt wird, sehen Sie in der Ranger-Admin-Benutzeroberfläche einen neuen Servicenamens „AMAZON-EMR-S3“, wie im folgenden Image gezeigt (Ranger-Version 2.0 wird angezeigt).



Schritt 4: Erstellen Sie eine Instanz der AMAZON-EMR-EMRFS Anwendung.

Erstellen Sie eine Instance der Servicedefinition.

- Klicken Sie auf das + neben AMAZON-EMR-EMRFS.

Füllen Sie die folgenden Felder aus:

Servicename (falls angezeigt): Der vorgeschlagene Wert ist **amazonemrs3**. Notieren Sie sich diesen Servicenamen, da er für die Erstellung einer EMR-Sicherheitskonfiguration benötigt wird.

Anzeigename: Der Name, der für diesen Service angezeigt wird. Der vorgeschlagene Wert ist **amazonemrs3**.

Allgemeiner Name für das Zertifikat: Das CN-Feld innerhalb des Zertifikats, das verwendet wird, um von einem Client-Plugin aus eine Verbindung zum Admin-Server herzustellen. Dieser Wert muss mit dem CN-Feld in Ihrem TLS-Zertifikat übereinstimmen, das für das Plugin erstellt wurde.

 **Access Manager** **Audit** **Security Zone** **Settings** admin

Service Manager > Edit Service

Edit Service

Service Details :

Service Name *

amazonemrs3

Display Name

amazonemrs3

Description

This is the EMRFS S3 Plugin.

Active Status

☒ Enabled ☐ Disabled

Select Tag Service

Select Tag Service

Config Properties :

Common Name for Certificate

CNOfCertificate

Add New Configurations

Name	Value

+

Test Connection

Save

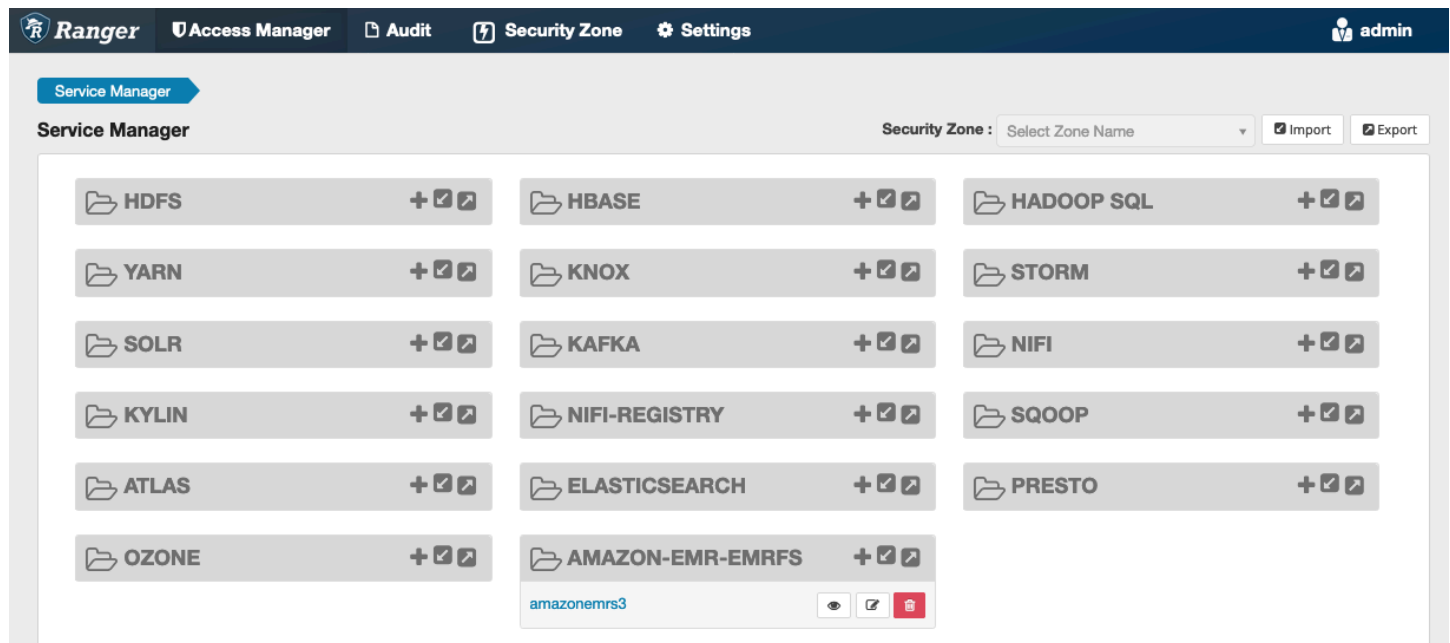
Cancel

Delete

Note

Das TLS-Zertifikat für dieses Plugin sollte im Trust-Store auf dem Ranger-Admin-Server registriert worden sein. Weitere Details finden Sie unter [TLS-Zertifikate für die Apache Ranger-Integration mit Amazon EMR](#).

Wenn der Service erstellt wird, enthält der Service Manager „AMAZON-EMR-EMRFS“, wie in der folgenden Abbildung dargestellt.



EMRFS-S3-Richtlinien erstellen

Füllen Sie die folgenden Felder aus, um auf der Seite Richtlinie erstellen des Service Managers eine neue Richtlinie zu erstellen.

Richtlinienname: Der Name dieser Richtlinie.

Richtlinienbezeichnung: Eine Bezeichnung, die Sie dieser Richtlinie hinzufügen können.

S3-Ressource: Eine Ressource, die mit dem Bucket und dem optionalen Präfix beginnt. Weitere Informationen finden Sie unter Bewährte Methoden für [Hinweise zur Verwendung von EMRFS-S3-Richtlinien](#). Ressourcen auf dem Ranger-Admin-Server sollten nicht **s3://**, **s3a://** oder **s3n://** enthalten.

Ranger Access Manager Audit Security Zone Settings admin

Service Manager amazonemr3 Policies Create Policy

Create Policy

Policy Details :

Policy Type **Access** [Add Validity Period](#)

Policy Name * **enabled** **normal**

Policy Label

S3 resource ***recursive**

Description

Audit Logging **YES**

Sie können Benutzer und Gruppen angeben, denen Berechtigungen erteilt werden sollen. Sie können auch Ausnahmen für Zulassungsbedingungen und Verweigerungsbedingungen angeben.

Audit Logging **YES**

Allow Conditions :

Select Role	Select Group	Select User		Delegate Admin
Select Roles	hadoop_analyst	analyst1	add/edit permissions ☒ GetObject ☒ PutObject ☒ ListObjects ☒ DeleteObject ☒ Select/Deselect All ☒ ☐ Add Permissions +	☐ ☒
+				

Deny All Other Accesses : **False**

Add

Cancel

 Note

Für jede Richtlinie sind maximal drei Ressourcen zulässig. Das Hinzufügen von mehr als drei Ressourcen kann zu einem Fehler führen, wenn diese Richtlinie auf einem EMR-Cluster verwendet wird. Beim Hinzufügen von mehr als drei Richtlinien wird eine Erinnerung an das Richtlinienlimit angezeigt.

Hinweise zur Verwendung von EMRFS-S3-Richtlinien

Bei der Erstellung von S3-Richtlinien in Apache Ranger sind einige Nutzungsaspekte zu beachten.

Berechtigungen für mehrere S3-Objekte

Sie können rekursive Richtlinien und Platzhalterausdrücke verwenden, um mehreren S3-Objekten mit gemeinsamen Präfixen Berechtigungen zu erteilen. Rekursive Richtlinien gewähren allen Objekten mit einem gemeinsamen Präfix Berechtigungen. Platzhalterausdrücke wählen mehrere Präfixe aus. Zusammen gewähren sie allen Objekten mit mehreren gemeinsamen Präfixen, wie in den folgenden Beispielen gezeigt.

Example Verwenden einer rekursiven Richtlinie

Angenommen, Sie benötigen Berechtigungen, um alle Parquet-Dateien in einem S3-Bucket aufzulisten, der wie folgt organisiert ist.

```
s3://sales-reports/americas/
+- year=2000
|   +- data-q1.parquet
|   +- data-q2.parquet
+- year=2019
|   +- data-q1.json
|   +- data-q2.json
|   +- data-q3.json
|   +- data-q4.json
|
+- year=2020
|   +- data-q1.parquet
|   +- data-q2.parquet
|   +- data-q3.parquet
|   +- data-q4.parquet
|   +- annual-summary.parquet
```

```
+ - year=2021
```

Betrachten Sie zunächst die Parquet-Dateien mit dem Präfix `s3://sales-reports/americas/year=2000`. Sie können allen auf zwei Arten `GetObject` Berechtigungen gewähren:

Verwenden von nichtrekursiven Richtlinien: Eine Option besteht darin, zwei separate nichtrekursive Richtlinien zu verwenden, eine für das Verzeichnis und die andere für die Dateien.

Die erste Richtlinie erteilt die Erlaubnis für das Präfix `s3://sales-reports/americas/year=2020` (es gibt keinen Trailing-/).

```
- S3 resource = "sales-reports/americas/year=2000"
- permission = "GetObject"
- user = "analyst"
```

Die zweite Richtlinie verwendet einen Platzhalterausdruck, um allen Dateien mit Präfix `sales-reports/americas/year=2020/` (beachten Sie das Trailing-/).

```
- S3 resource = "sales-reports/americas/year=2020/*"
- permission = "GetObject"
- user = "analyst"
```

Verwendung einer rekursiven Richtlinie: Eine bequemere Alternative besteht darin, eine einzige rekursive Richtlinie zu verwenden und dem Präfix rekursive Berechtigungen zu erteilen.

```
- S3 resource = "sales-reports/americas/year=2020"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Bisher waren nur die Parquet-Dateien mit dem Präfix `s3://sales-reports/americas/year=2000` enthalten. Sie können jetzt auch die Parquet-Dateien mit einem anderen Präfix, `s3://sales-reports/americas/year=2020`, in dieselben rekursive Richtlinie aufnehmen, indem Sie einen Platzhalterausdruck wie folgt einfügen.

```
- S3 resource = "sales-reports/americas/year=20?0"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Richtlinien für PutObject und DeleteObject Berechtigungen

Das Schreiben von Richtlinien PutObject und DeleteObject Berechtigungen für Dateien auf EMRFS erfordert besondere Sorgfalt, da sie im Gegensatz zu GetObject Berechtigungen zusätzliche rekursive Berechtigungen erfordern, die dem Präfix gewährt werden.

Example Richtlinien für und Berechtigungen PutObject DeleteObject

Zum Löschen der Datei ist beispielsweise nicht nur eine DeleteObject Berechtigung für die eigentliche Datei `annual-summary.parquet` erforderlich.

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "DeleteObject"
- user = "analyst"
```

Außerdem ist eine Richtlinie erforderlich, die rekursive Rechte GetObject und PutObject Berechtigungen für das zugehörige Präfix gewährt.

In ähnlicher Weise erfordert das Ändern der Datei `annual-summary.parquet` nicht nur eine PutObject-Berechtigung für die eigentliche Datei.

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "PutObject"
- user = "analyst"
```

Außerdem ist eine Richtlinie erforderlich, die eine rekursive GetObject-Erlaubnis für ihr Präfix erteilt.

```
- S3 resource = "sales-reports/americas/year=2020"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Platzhalter in Richtlinien

Es gibt zwei Bereiche, in denen Platzhalter angegeben werden können. Bei der Angabe einer S3-Ressource können „*“ und „?“ verwendet werden. Das „*“ ermöglicht einen Abgleich mit einem S3-Pfad und entspricht allem, was hinter dem Präfix steht. Zum Beispiel die folgende Richtlinie.

```
S3 resource = "sales-reports/americas/*"
```

Dies entspricht den folgenden S3-Pfaden.

```
sales-reports/americas/year=2020/  
sales-reports/americas/year=2019/  
sales-reports/americas/year=2019/month=12/day=1/afile.parquet  
sales-reports/americas/year=2018/month=6/day=1/afile.parquet  
sales-reports/americas/year=2017/afile.parquet
```

Das Platzhalterzeichen „?“ entspricht nur einem einzelnen Zeichen. Beispielsweise für die Richtlinie.

```
S3 resource = "sales-reports/americas/year=201?/"
```

Dies entspricht den folgenden S3-Pfaden.

```
sales-reports/americas/year=2019/  
sales-reports/americas/year=2018/  
sales-reports/americas/year=2017/
```

Platzhalter bei Benutzern

Bei der Zuweisung von Benutzern, die Benutzern Zugriff gewähren sollen, gibt es zwei integrierte Platzhalter. Der erste ist der Platzhalter „{USER}“, der allen Benutzern Zugriff gewährt. Der zweite Platzhalter ist „{OWNER}“, der Zugriff auf den Eigentümer eines bestimmten Objekts oder direkt ermöglicht. Der Platzhalter „{USER}“ wird derzeit jedoch nicht unterstützt.

Einschränkungen

Die folgenden Einschränkungen gelten derzeit für das EMRFS S3-Plugin:

- Apache-Ranger-Richtlinien können maximal drei Richtlinien haben.
- Der Zugriff auf S3 muss über EMRFS erfolgen und kann mit Hadoop-bezogenen Anwendungen verwendet werden. Folgendes wird nicht unterstützt:
 - Boto3-Bibliotheken
 - AWS SDK und AWK CLI
 - S3A-Open-Source-Konnektor
- Die Ablehnungs-Richtlinien von Apache Ranger werden nicht unterstützt.
- Vorgänge auf S3 mit Schlüsseln mit CSE-KMS-Verschlüsselung werden derzeit nicht unterstützt.

- Die Freigabe über Regionsgrenzen hinweg wird nicht unterstützt.
- Das Sicherheitszone-Feature von Apache Ranger wird nicht unterstützt. Einschränkungen der Zugriffskontrolle, die mit demr Sicherheitszone-Feature definiert wurden, gelten nicht für Ihre Amazon-EMR-Cluster.
- Der Hadoop-Benutzer generiert keine Prüfereignisse, da Hadoop immer auf das Instanzprofil zugreift. EC2
- Es wird empfohlen, Amazon EMR Consistency View zu deaktivieren. S3 ist stark konsistent und wird daher nicht mehr benötigt. Weitere Informationen finden Sie unter [Starke Konsistenz von Amazon-S3](#).
- Das EMRFS-S3-Plugin führt zahlreiche STS-Aufrufe durch. Es wird empfohlen, Lasttests auf einem Entwicklungskonto durchzuführen und das STS-Aufrufvolumen zu überwachen. Es wird außerdem empfohlen, eine STS-Anfrage zu stellen, um die Dienstlimits anzuheben AssumeRole.
- Der Ranger Admin-Server unterstützt die automatische Vervollständigung nicht.

Trino-Plugin für die Ranger-Integration mit Amazon EMR

Trino (früher PrestoSQL) ist eine SQL-Abfrage-Engine, mit der Sie Abfragen für Datenquellen wie HDFS, Objektspeicher, relationale Datenbanken und NoSQL-Datenbanken ausführen können. Es macht die Migration von Daten an einen zentralen Ort überflüssig und ermöglicht es Ihnen, die Daten von jedem Ort aus abzufragen. Amazon EMR bietet ein Apache- Ranger-Plugin für detaillierte Zugriffskontrollen für Trino. Das Plugin ist mit Admin-Server-Version von Open-Source-Apache-Ranger 2.0 und höher kompatibel.

Themen

- [Unterstützte Features](#)
- [Installation der Servicekonfiguration](#)
- [Erstellen von Trino-Richtlinien](#)
- [Überlegungen](#)
- [Einschränkungen](#)

Unterstützte Features

Das Apache-Ranger-Plugin für Trino auf Amazon EMR unterstützt die gesamte Funktionalität der Trino-Abfrage-Engine, die durch eine detaillierte Zugriffskontrolle geschützt ist. Dazu gehören Zugriffskontrollen auf Datenbank-, Tabellen- und Spaltenebene sowie Zeilenfilterung und

Datenmaskierung. Die Richtlinien von Apache Ranger können Richtlinien für die Gewährung und die Ablehnung von Benutzern und Gruppen beinhalten. Prüfereignisse werden auch in CloudWatch Protokolle aufgenommen.

Installation der Servicekonfiguration

Die Installation der Trino-Servicedefinition erfordert die Einrichtung des Ranger-Admin-Servers. Informationen zum Einrichten des Ranger-Admin-Servers finden Sie unter [Richten Sie einen Ranger Admin-Server für die Integration mit Amazon EMR ein](#).

Zum Installieren der Trino-Servicedefinition führen Sie die folgenden Schritte aus.

1. Stellen Sie eine SSH-Verbindung zum Apache-Ranger-Admin-Server her.

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

2. Deinstallieren Sie das Presto-Server-Plugin, falls es existiert. Führen Sie den folgenden Befehl aus. Wenn dieser Fehler mit der Fehlermeldung „Service nicht gefunden“ angezeigt wird, bedeutet dies, dass das Presto-Server-Plugin nicht auf Ihrem Server installiert wurde. Fahren Sie mit dem nächsten Schritt fort.

```
curl -f -u *<admin users login>:*:<_**_password_ **_for_** _ranger admin  
user_**_>_* -X DELETE -k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/  
v2/api/servicedef/name/presto'
```

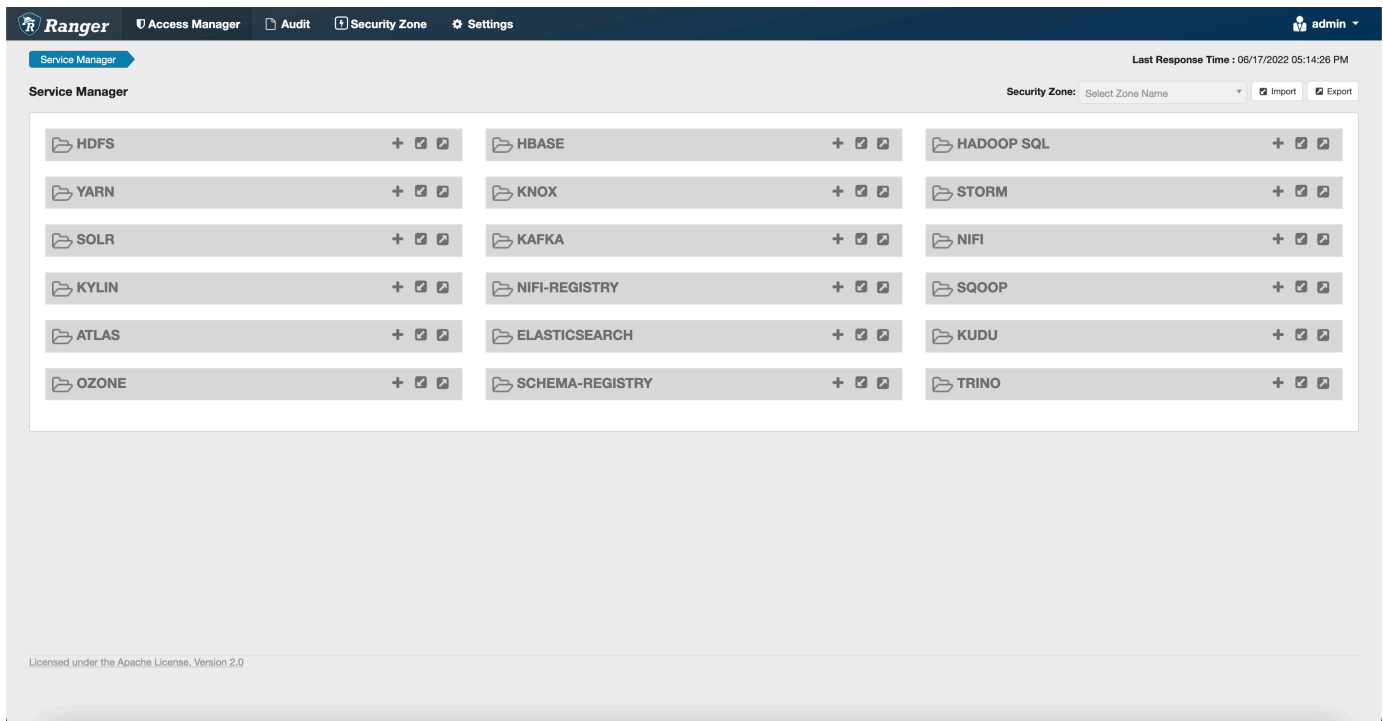
3. Laden Sie die Servicedefinition und das Apache-Ranger-Admin-Server-Plugin herunter. Laden Sie die Servicedefinition in einem temporären Verzeichnis herunter. Diese Servicedefinition wird von Ranger-2.x-Versionen unterstützt.

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/  
version-2.0/ranger-servicedef-amazon-emr-trino.json
```

4. Registrieren Sie die Apache-Spark-Servicedefinition für Amazon EMR

```
curl -u *<admin users login>:*:<_**_password_ **_for_** _ranger admin user_**_>_*  
-X POST -d @ranger-servicedef-amazon-emr-trino.json \  
-H "Accept: application/json" \  
-H "Content-Type: application/json" \  
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

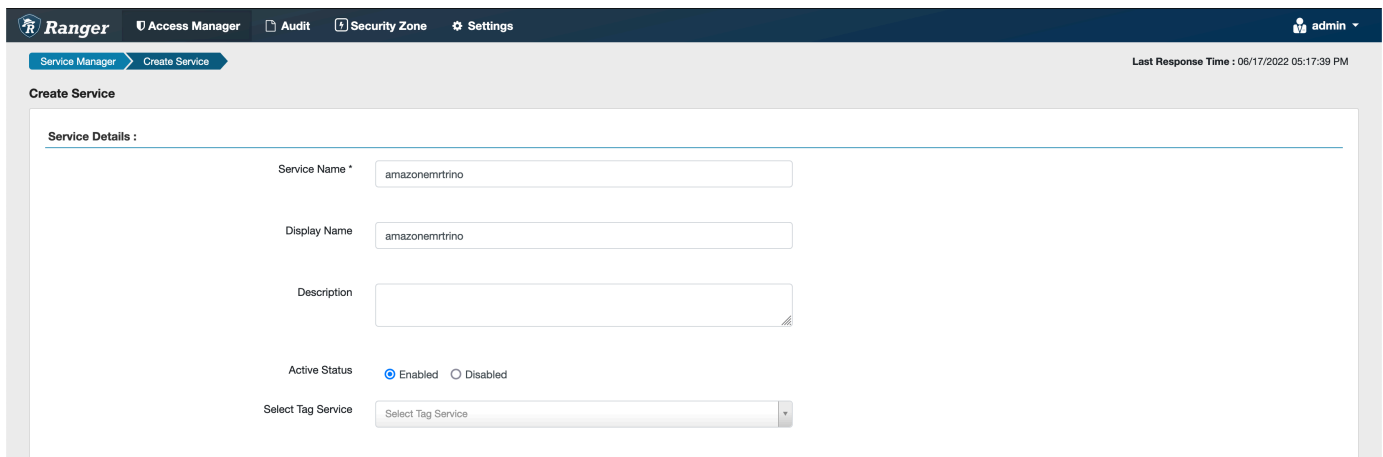
Wenn dieser Befehl erfolgreich ausgeführt wird, wird in Ihrer Ranger-Admin-Benutzeroberfläche ein neuer Service mit dem Namen TRINO angezeigt, wie in der folgenden Abbildung dargestellt.



- Erstellen Sie eine Instance der TRINO-Anwendung und geben Sie die folgenden Informationen ein.

Servicename: Der Servicename, den Sie verwenden werden. Der vorgeschlagene Wert ist `amazonemrtrino`. Notieren Sie sich diesen Servicennamen, da er für die Erstellung einer Amazon-EMR-Sicherheitskonfiguration benötigt wird.

Anzeigename: Der Name, der für diese Instance angezeigt werden soll. Der vorgeschlagene Wert ist `amazonemrtrino`.



jdbc.driver. ClassName: Der Klassenname der JDBC-Klasse für Trino-Konnektivität. Sie können den Standardwert verwenden.

jdbc.url: Die JDBC-Verbindungszeichenfolge, die verwendet werden soll, wenn eine Verbindung zum Trino-Koordinator hergestellt wird.

Allgemeiner Name für das Zertifikat: Das CN-Feld innerhalb des Zertifikats, das verwendet wird, um von einem Client-Plugin aus eine Verbindung zum Admin-Server herzustellen. Dieser Wert muss mit dem CN-Feld in Ihrem TLS-Zertifikat übereinstimmen, das für das Plugin erstellt wurde.

Das TLS-Zertifikat für dieses Plugin sollte im Trust-Store auf dem Ranger-Admin-Server registriert worden sein. Weitere Informationen finden Sie unter [TLS-Zertifikate](#).

Erstellen von Trino-Richtlinien

Wenn Sie eine neue Richtlinie erstellen, füllen Sie die folgenden Felder aus.

Richtliniennamen: Der Name dieser Richtlinie.

Richtlinienbezeichnung: Eine Bezeichnung, die Sie dieser Richtlinie hinzufügen können.

Tabelle: Die Tabellen, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Tabellen.

Schema: Die Schemas, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Schemas.

Tabelle: Die Tabellen, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Tabellen.

Spalte: Die Spalten, für die diese Richtlinie gilt. Der Platzhalter „*“ steht für alle Spalten.

Beschreibung: Eine Beschreibung dieser Richtlinie.

Andere Arten von Richtlinien gibt es für den Trino-Benutzer (für den Zugriff, der sich als Benutzer ausgibt), die Trino-System-/Sitzungseigenschaft (für die Änderung der System- oder Sitzungseigenschaften der Engine), für Funktionen/Prozeduren (für das Zulassen von Funktions- oder Prozeduraufrufen) und die URL (für die Gewährung von Lese-/Schreibzugriff auf die Engine an Datenspeicherorten).

The screenshot shows the 'Create Policy' page in the Apache Ranger web interface. The top navigation bar includes 'Ranger', 'Access Manager', 'Audit', 'Security Zone', and 'Settings'. The user is logged in as 'admin'. The breadcrumb trail is 'Service Manager > amazonemrtrino Policies > Create Policy'. The 'Last Response Time' is 06/17/2022 05:22:12 PM.

The 'Policy Details' section contains the following fields and controls:

- Policy Type:** Set to 'Access'.
- Policy Name:** A text input field containing 'policyName'.
- Policy Label:** A text input field containing 'Policy Label'.
- Enabled/Normal:** Two radio buttons. 'Enabled' is selected.
- Resources and Permissions:**
 - catalog:** Set to 'hive'. Permission: 'Include' (selected).
 - schema:** Set to '*'. Permission: 'Include' (selected).
 - table:** Set to '*'. Permission: 'Include' (selected).
 - column:** Set to '*'. Permission: 'Include' (selected).
- Description:** A large text area for providing a description.
- Audit Logging:** A checkbox labeled 'Yes'.

A button labeled 'Add Validity Period' is located in the top right corner of the form.

Um die Benutzer und Gruppen anzugeben, geben Sie die Benutzer und Gruppen unten ein, um Berechtigungen zu erteilen. Sie können auch Ausnahmen für Zulassungsbedingungen und Verweigerungsbedingungen angeben.

Allow Conditions:

Select Role	Select Group	Select User	Permissions	Delegate Admin
Select Roles	public	(USER)	Add Permissions	
+ Exclude from Allow Conditions:				
Select Roles	Select Groups	Select Users	Add Permissions	
+ Exclude from Deny Conditions:				

Deny All Other Accesses: ☐ False

Deny Conditions:

Select Role	Select Group	Select User	Permissions	Delegate Admin
Select Roles	Select Groups	Select Users	Add Permissions +	
+ Exclude from Deny Conditions:				

javascript: Select Role Select Group Select User Permissions Delegate Admin

Nachdem Sie die Bedingungen für das Zulassen und Verweigern angegeben haben, klicken Sie auf Speichern.

Überlegungen

Bei der Erstellung von Trino-Richtlinien in Apache Ranger sind einige Nutzungsaspekte zu beachten.

Hive-Metadatenserver

Auf den Hive-Metadatenserver können nur vertrauenswürdige Engines zugreifen, insbesondere die Trino-Engine, um sich vor unbefugtem Zugriff zu schützen. Auf den Hive-Metadatenserver greifen auch alle Knoten im Cluster zu. Der erforderliche Port 9 083 ermöglicht allen Knoten den Zugriff auf den Hauptknoten.

Authentifizierung

Standardmäßig ist Trino so konfiguriert, dass es sich mithilfe von Kerberos authentifiziert, wie in der Amazon-EMR-Sicherheitskonfiguration konfiguriert.

Verschlüsselung während der Übertragung erforderlich

Für das Trino-Plugin müssen Sie die Verschlüsselung während der Übertragung in der Amazon-EMR-Sicherheitskonfiguration aktiviert haben. Informationen zum Aktivieren der Verschlüsselung finden Sie unter [Verschlüsselung während der Übertragung](#).

Einschränkungen

Im Folgenden sind die aktuellen Einschränkungen des Trino-Plugins aufgeführt:

- Der Ranger-Admin-Server unterstützt die automatische Vervollständigung nicht.

Fehlerbehebung für Apache Ranger

Im Folgenden finden Sie einige häufig diagnostizierte Probleme im Zusammenhang mit der Verwendung von Apache Ranger.

Empfehlungen

- Testen Sie mit einem einzigen Hauptknotencluster: Master-Cluster mit einem Knoten können schneller bereitgestellt werden als ein Cluster mit mehreren Knoten, wodurch die Zeit für jede Test-Iteration verkürzt werden kann.
- Stellen Sie den Entwicklungsmodus auf dem Cluster ein. Wenn Sie Ihren EMR-Cluster starten, setzen Sie den `--additional-info`-Parameter auf:

```
'{"clusterType":"development"}'
```

Dieser Parameter kann nur über die AWS CLI oder das AWS SDK festgelegt werden und ist nicht über die Amazon EMR-Konsole verfügbar. Wenn dieses Flag gesetzt ist und der Master nicht bereitstellen kann, hält der Amazon-EMR-Service den Cluster für einige Zeit am Leben, bevor er außer Betrieb genommen wird. Diese Zeit ist sehr nützlich, um verschiedene Protokolldateien zu überprüfen, bevor der Cluster beendet wird.

EMR-Cluster konnte nicht bereitgestellt werden

Es gibt mehrere Gründe, warum ein Amazon-EMR-Cluster möglicherweise nicht gestartet werden kann. Im Folgenden finden Sie einige Möglichkeiten, das Problem zu diagnostizieren.

Überprüfen Sie die EMR-Bereitstellungsprotokolle

Amazon EMR verwendet Puppet, um Anwendungen auf einem Cluster zu installieren und zu konfigurieren. Anhand der Protokolle erhalten Sie Informationen darüber, ob während der Bereitstellungsphase eines Clusters Fehler aufgetreten sind. Auf die Protokolle kann auf dem Cluster oder in S3 zugegriffen werden, wenn die Protokolle so konfiguriert sind, dass sie an S3 übertragen werden.

Die Protokolle werden auf der Festplatte `/var/log/provision-node/apps-phase/0/{UUID}/puppet.log` und `s3://<LOG_LOCATION>/<CLUSTER_ID>/node/<EC2_INSTANCE_ID>/provision-node/apps-phase/0/{UUID}/puppet.log.gz` gespeichert.

Allgemeine Fehlermeldungen

Fehlermeldung	Ursache
Puppet (err): Der Systemd-Start für ist fehlgeschlagen! emr-record-server journalctl-Protokoll für: emr-record-server	EMR Record Server konnte nicht gestartet werden. Weitere Informationen finden Sie unten in den EMR-Record-Server-Protokollen.
Puppet (err): Der Systemd-Start für ist fehlgeschlagen! emr-record-server journalctl-Protokoll für emrsecretagent:	EMR Record Server konnte nicht gestartet werden. Weitere Informationen finden Sie weiter unten unter Secret-Agent-Protokolle überprüfen.
/Stage [main] /Ranger_Plugins: :Ranger_Hive_ (Hinweis): 140408606197664:Fehler:0906d06c:PEM-Routinen:PEM_READ_BIO:Keine Startzeile:PEM_LIB.C:707:Erwartet: JEDER PRIVATE SCHLÜSSEL plugin/Ranger_plugins::Prepare_two_way_tls[configure 2-way TLS in Hive plugin]/Exec[create keystore and truststore for Ranger Hive plugin]/returns	Das private TLS-Zertifikat in Secret Manager für das Apache Ranger-Plug-in-Zertifikat hat nicht das richtige Format oder ist kein privates Zertifikat. Informationen zu TLS-Zertifikate für die Apache Ranger-Integration mit Amazon EMR den Zertifikatsformaten finden Sie unter.
/Stage [main] /Ranger_Plugins: :Ranger_S3_plugin/Ranger_plugins::Prepare_two_way_tls[configure 2-way TLS in Ranger s3 plugin]/Exec[create keystore and truststore for Ranger amazon-emr-s3 plugin]/returns (notice): An error occurred (AccessDeniedException) when calling the GetSecretValue operation: User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/EMR_EC2_DefaultRole/i -XXXXXXXXXXXX	Die EC2 Instanzprofilrolle hat nicht die richtigen Berechtigungen, um die TLS-Zertifikate von Secrets Agent abzurufen.

Fehlermeldung	Ursache
ist nicht berechtigt, Folgendes auszuführen: secretsmanager: GetSecretValue on resource: arn:aws:secretsmanager:us-east-1:xxx xxxxxxx:secret: -XXXXX AdminServer	

Überprüfen Sie die SecretAgent Protokolle

Secret-Agent-Protokolle befinden sich in `/emr/secretagent/log/` auf einem EMR-Knoten oder im `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/daemons/secretagent/-`Verzeichnis in S3.

Allgemeine Fehlermeldungen

Fehlermeldung	Ursache
Ausnahme im Thread „main“ <code>com.amazonaws.services.securitytoken.model.AWSSecurityTokenServiceException: Benutzer: arn:aws:sts::XXXXXXXXXXXX:assumed-role/EMR_EC2_DefaultRole/i-XXXXXXXXXXXX ist nicht berechtigt, Folgendes auszuführen: sts: AssumeRole on resource: arn:aws:iam::XXXXXXXXXXXX:role/* (Service: AWSSecurityTokenService; Statuscode: 403; Fehlercode:; Fehlercode:; Anforderungs-ID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX; Proxy: null) AccessDenied</code>	Die obige Ausnahme bedeutet, dass die EC2 EMR-Instanzprofilrolle nicht berechtigt ist, die Rolle <code>RangerPluginDataAccessRole</code> zu übernehmen. Siehe IAM-Rollen für die native Integration mit Apache Ranger .
FEHLER <code>qtp54617902-149: Eine Web-App-Ausnahme ist aufgetreten</code> <code>javax.ws.rs.NotAllowedException: HTTP 405-Methode nicht erlaubt</code>	Diese Fehler können ignoriert werden.

Überprüfen Sie die Record-Serverprotokolle (für SparkSQL)

EMR Record Server-Protokolle sind unter `/var/log/emr-record-server/` auf einem EMR-Knoten oder im Verzeichnis `s3:///<LOG LOCATION><CLUSTER ID>/node/ < INSTANCE ID>/daemons//` in S3 verfügbar. EC2 `emr-record-server`

Allgemeine Fehlermeldungen

Fehlermeldung	Ursache
InstanceMetadataServiceResourceFetcherEMR Record Server-Logs sind unter <code>/-record-server/</code> auf einem EMR-Knoten oder im Verzeichnis <code>s3:///node/ < INSTANCE ID>/daemons//</code> in S3 verfügbar. -----sep-----:105 - [] Token com.amazonaws konnte nicht abgerufen werden. SdkClientException: Es konnte keine Verbindung zum Service-Endpunkt hergestellt werden	Das EMR SecretAgent wurde nicht angezeigt oder es liegt ein Problem vor. Untersuchen Sie die SecretAgent Protokolle auf Fehler und überprüfen Sie das Puppet-Skript, um festzustellen, ob Bereitstellungsfehler aufgetreten sind.

Abfragen für die Ranger-Integration mit Amazon EMR schlagen unerwartet fehl

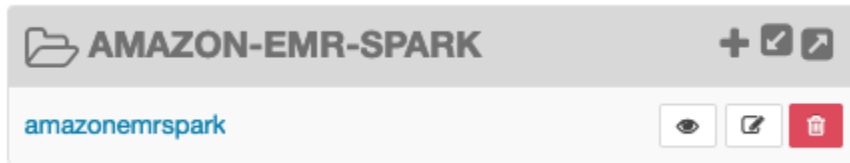
Überprüfen Sie die Protokolle des Apache Ranger-Plug-ins (Apache Hive, EMRRecordServer, EMR usw. SecretAgent, Protokolle)

Dieser Abschnitt gilt für alle Anwendungen, die in das Ranger-Plugin integriert sind, wie Apache Hive, EMR Record Server und EMR. SecretAgent

Allgemeine Fehlermeldungen

Fehlermeldung	Ursache
ERROR:272 PolicyRefresher - [] (PolicyRefresherserviceName=policy-repository): Der Dienst konnte nicht gefunden werden. Löscht den lokalen Richtliniencache (-1)	Diese Fehlermeldungen bedeuten, dass der Servicename, den Sie in der EMR-Sicherheitskonfiguration angegeben haben, nicht mit einem Service-Richtlinien-Repository auf dem Ranger-Admin-Server übereinstimmt.

Wenn Ihr AMAZON-EMR-SPARK Dienst auf dem Ranger Admin-Server wie folgt aussieht, sollten Sie ihn als Dienstnamen eingeben. **amazonemrspark**



Arbeiten mit AWS Glue-Datenkatalogansichten in Amazon EMR (Vorschau)

Important

AWS Glue Data Catalog-Ansichten in Amazon EMR on EC2 befinden sich in der Vorschauversion und die Funktion kann sich ändern. Die Funktion wird in der Vorschauversion gemäß der Definition in den [AWS Servicebedingungen](#) bereitgestellt. AWS Glue Data Catalog-Ansichten haben die allgemeine Verfügbarkeit für EMR Serverless erreicht. Weitere Informationen finden Sie unter [Arbeiten mit Glue-Datenkatalogansichten](#) im Amazon EMR Serverless User Guide.

Sie können einzelne gemeinsame Ansichten im AWS Glue-Datenkatalog erstellen und verwalten. Einzelne allgemeine Ansichten sind nützlich, da sie mehrere SQL-Abfrage-Engines unterstützen, sodass Sie auf dieselbe Ansicht in verschiedenen Ansichten zugreifen können AWS-Services, z. B. in Amazon EMR, Amazon Athena und Amazon Redshift.

Indem Sie eine Ansicht im Datenkatalog erstellen, können Sie Ressourcenzuweisungen und tagbasierte Zugriffskontrollen verwenden, um Zugriff auf eine Datenkatalogansicht AWS Lake Formation zu gewähren. Mit dieser Methode der Zugriffskontrolle müssen Sie keinen zusätzlichen Zugriff auf die Tabellen konfigurieren, auf die Sie beim Erstellen der Ansicht verwiesen haben. Diese Methode zur Erteilung von Berechtigungen wird Definer-Semantik genannt, und diese Ansichten werden Defineransichten genannt. Weitere Informationen zur Zugriffskontrolle in Lake Formation finden Sie unter [Erteilen und Widerrufen von Berechtigungen für Datenkatalogressourcen](#). im AWS Lake Formation Entwicklerhandbuch.

Datenkatalogansichten sind für die folgenden Anwendungsfälle nützlich:

- Granulare Zugriffskontrolle — Erstellen Sie eine Ansicht, die den Datenzugriff auf der Grundlage der vom Benutzer benötigten Berechtigungen einschränkt. Mithilfe von Ansichten im Datenkatalog können Sie beispielsweise verhindern, dass Mitarbeiter, die nicht in der Personalabteilung arbeiten, persönlich identifizierbare Informationen (PII) sehen.
- Vollständige Definition der Ansicht — Indem Sie bestimmte Filter auf Ihre Ansicht im Datenkatalog anwenden, stellen Sie sicher, dass die Datensätze in einer Ansicht im Datenkatalog immer vollständig sind.
- Verbesserte Sicherheit — Die zur Erstellung der Ansicht verwendete Abfragedefinition muss vollständig sein. Dieser Vorteil bedeutet, dass Ansichten im Datenkatalog weniger anfällig für SQL-Befehle von böswilligen Spielern sind.
- Einfaches Teilen von Daten — teilen Sie Daten mit anderen, AWS-Konten ohne Daten zu verschieben. Weitere Informationen finden Sie unter [Kontoübergreifender Datenaustausch in Lake Formation](#).

Erstellen einer Data-Catalog-Ansicht

Important

In dieser Vorschauversion validiert Amazon EMR das Spark-SQL, das Sie beim Erstellen der Ansicht verwenden, nicht. Um Risiken zu verringern, empfehlen wir, die Anzahl der Benutzer zu beschränken, denen Sie Berechtigungen zum Erstellen von Ansichten gewähren.

Um eine Datenkatalogansicht zu erstellen, müssen Sie eine IAM-Rolle verwenden, die über die vollen SELECT Berechtigungen mit `Grantable` Optionen für alle Tabellen verfügt, auf die Sie beim Erstellen der Ansicht verweisen möchten. Diese Rolle wird als Definiererrolle bezeichnet. Eine vollständige Liste der Berechtigungen und Voraussetzungen, die zum Erstellen einer Datenkatalogansicht erforderlich sind, finden Sie unter [Arbeiten mit Ansichten](#) im AWS Lake Formation Entwicklerhandbuch. Sie müssen die verwendete AWS CLI , um Ihre IAM-Rolle zu konfigurieren. Weitere Informationen finden [Sie unter Verwenden einer IAM-Rolle in der AWS CLI](#).

Gehen Sie wie folgt vor, um eine Datenkatalogansicht zu erstellen.

Note

Um von Apache Spark auf Amazon EMR auf eine Datenkatalogansicht zuzugreifen, müssen Sie den Dialekt auf SPARK und auf einstellen. DialectVersion 3.4.1-amzn-2

1. Laden Sie zuerst das Vorschaumodell herunter.

```
aws s3 cp s3://emr-data-access-control-us-east-1/beta/glue-views/model/
service-2.json
```

2. Konfigurieren Sie das AWS CLI , um das Vorschaumodell zu verwenden.

```
aws configure add-model --service-model file:///<path-to-preview-model>/
service-2.json --service-name glue-views
```

3. Erstellen Sie die Ansicht.

```
aws glue-views create-table --cli-input-json '{
  "DatabaseName": "<database>",
  "TableInput": {
    "Name": "<view>",
    "StorageDescriptor": {
      "Columns": [
        {
          "Name": "<col1>",
          "Type": "<data-type>"
        },
        ...
        {
          "Name": "<colN>",
          "Type": "<data-type>"
        }
      ]
    },
    "ViewDefinition": {
      "SubObjects": [
        "arn:aws:glue:<aws-region>:<aws-account-id>:table/<database>/<referenced-
table1>",
        ...
        "arn:aws:glue:<aws-region>:<aws-account-id>:table/<database>/<referenced-
tableN>",

```

```

    ],
    "IsProtected": true,
    "Representations": [
      {
        "Dialect": "SPARK",
        "DialectVersion": "3.4.1-amzn-2",
        "ViewOriginalText": "<Spark-SQL>",
        "ViewExpandedText": "<Spark-SQL>"
      }
    ]
  }
}
}'

```

Zugriff auf eine Datenkatalogansicht aktivieren

Important

Es wird empfohlen, den Zugriff auf Datenkatalogansichten nur mit EMR-Clustern in Testumgebungen und nicht in Produktionsumgebungen zu aktivieren.

Um von Apache Spark auf Amazon EMR auf die Datenkatalogansicht zuzugreifen, müssen Sie zuerst die Unterstützung für Lake Formation aktivieren und das folgende Skript verwenden, um die Unterstützung für Ansichten mit Spark auf Amazon EMR zu aktivieren. Weitere Informationen zur Aktivierung der Unterstützung finden Sie unter [Aktivieren von Lake Formation mit Amazon EMR](#) und [Verwenden benutzerdefinierter Bootstrap-Aktionen](#).

```

# Download the script and upload it to Amazon S3
wget https://emr-data-access-control-us-east-1.s3.amazonaws.com/beta/glue-views/ba/
enable-mdv.sh /Users/$USER/enable-mdv.sh
aws s3 cp /Users/$USER/enable-views.sh s3://<bucket>/<prefix>/enable-views.sh

# EMR Security Configuration
cat <<EOT > /Users/$USER/lakeformation-protection.json
{
  "AuthorizationConfiguration":{
    "IAMConfiguration":{

```

```

        "EnableApplicationScopedIAMRole":true
    },
    "LakeFormationConfiguration":{
        "AuthorizedSessionTagValue":"Amazon EMR"
    }
},
"EncryptionConfiguration": {
    "EnableInTransitEncryption": true,
    "InTransitEncryptionConfiguration": {
        "TLSCertificateConfiguration": {
            "CertificateProviderType": "PEM",
            "S3Object": "s3://<BUCKET>/<PREFIX>/certificates.zip"
        }
    }
}
}
EOT

```

```
SECURITY_CONFIG="RuntimeRolesWithAWSLakeFormation"
```

```
aws emr create-security-configuration \
--name $SECURITY_CONFIG \
--security-configuration file:///Users/$USER/lakeformation-protection.json
```

```
# EMR Cluster version
RELEASE_LABEL="emr-6.15.0"
```

Verwenden Sie dann den folgenden AWS CLI Befehl, der die Bootstrap-Aktion verwendet, um einen EMR-Cluster zu erstellen, der Datenkatalogansichten unterstützt.

```
aws emr create-cluster \
...
--release-label $RELEASE_LABEL \
--security-configuration $SECURITY_CONFIG \
--bootstrap-actions \
  Name='Enable Views',Path="s3://<bucket>/<prefix>/enable-views.sh"
```

Abfrage einer Data-Catalog-Ansicht

Important

In dieser Vorschauversion empfehlen wir, dass Sie nur auf Ansichten aus vertrauenswürdigen Quellen zugreifen. In der Vorschauversion verfügt Amazon EMR über eine begrenzte Anzahl von Validierungen, die Ihren EMR-Cluster schützen.

Nachdem Sie eine Datenkatalog-Ansicht erstellt haben, können Sie nun eine IAM-Rolle verwenden, um die Ansicht abzufragen. Die IAM-Rolle muss über die SELECT Berechtigung für die Datenkatalogansicht verfügen. Sie müssen keinen Zugriff auf die zugrunde liegenden Tabellen gewähren, auf die in der Ansicht verwiesen wird. Sie müssen diese IAM-Rolle als Runtime-Rolle verwenden. Sie können von einem EMR-Cluster aus mithilfe einer Runtime-Rolle aus Amazon EMR Steps, EMR Studio und SageMaker AI Studio auf die Ansicht zugreifen. Weitere Informationen zu Runtime-Rollen finden Sie unter [Schritte zu Runtime-Rollen für Amazon EMR](#).

Sobald Sie alles eingerichtet haben, können Sie Ihre Ansicht abfragen. Nachdem Sie beispielsweise den EMR-Cluster an Ihren Workspace in EMR Studio angehängt haben, können Sie die folgende Abfrage ausführen, um auf eine Ansicht zuzugreifen.

```
SELECT * from <database>.<glue-data-catalog-view> LIMIT 10
```

Einschränkungen

Beachten Sie die folgenden Einschränkungen, wenn Sie Datenkatalog-Ansichten verwenden.

- Sie können Datenkatalogansichten nur mit Amazon EMR 6.15.0 erstellen.
- Sie können in der Ansichtsdefinition nur auf bis zu 10 Tabellen verweisen.
- Sie können nur PROTECTED Datenkatalogsichten erstellen. UNPROTECTED Ansichten werden nicht unterstützt.
- In Datenkatalogansichten können Sie nicht auf Tabellen AWS-Konto in anderen Tabellen verweisen.
- Benutzerdefinierte Funktionen (UDFs) werden nicht unterstützt.
- In Datenkatalogansichten können Sie nicht auf offene Tabellenformate wie Apache Hudi oder Apache Iceberg verweisen.

- Sie können in Datenkatalogansichten nicht auf andere Ansichten verweisen.

Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster

Sicherheitsgruppen dienen als virtuelle Firewalls für EC2 Instances in Ihrem Cluster, um den eingehenden und ausgehenden Datenverkehr zu kontrollieren. Für jede Sicherheitsgruppe gibt es einen Satz von Regeln zur Kontrolle des eingehenden Datenverkehrs und einen Satz von Regeln zur Kontrolle des ausgehenden Datenverkehrs. Weitere Informationen finden Sie unter [EC2 Amazon-Sicherheitsgruppen für Linux-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Sie verwenden zwei Klassen von Sicherheitsgruppen mit Amazon EMR: Amazon-EMR-verwaltete Sicherheitsgruppen und zusätzliche Sicherheitsgruppen.

Mit jedem Cluster sind verwaltete Sicherheitsgruppen verknüpft. Sie können die standardmäßigen verwalteten Sicherheitsgruppen die Amazon EMR erstellt oder benutzerdefinierte verwaltete Sicherheitsgruppen angeben. In beiden Fällen fügt Amazon EMR automatisch Regeln zu verwalteten Sicherheitsgruppen hinzu, die ein Cluster für die Kommunikation zwischen Cluster-Instances und AWS Services benötigt.

Zusätzliche Sicherheitsgruppen sind optional. Sie können sie zusätzlich zu den verwalteten Sicherheitsgruppen angeben, um den Zugriff auf Cluster-Instances anzupassen. Zusätzliche Sicherheitsgruppen enthalten nur von Ihnen definierte Regeln. Amazon EMR ändert sie nicht.

Die von Amazon EMR in verwalteten Sicherheitsgruppen erstellten Regeln gestatten dem Cluster nur die Kommunikation zwischen internen Komponenten. Um Benutzern und Anwendungen den Zugriff auf einen Cluster von außerhalb des Clusters zu ermöglichen, können Sie Regeln in verwalteten Sicherheitsgruppen bearbeiten, zusätzliche Sicherheitsgruppen mit zusätzlichen Regeln erstellen oder beides ausführen.

Important

Das Bearbeiten von Regeln in verwalteten Sicherheitsgruppen kann unbeabsichtigte Folgen haben. Möglicherweise blockieren Sie versehentlich den Datenverkehr, der für die ordnungsgemäße Funktion der Cluster erforderlich ist, und verursachen Fehler, da die Knoten nicht erreichbar sind. Planen und testen Sie Sicherheitsgruppenkonfigurationen sorgfältig, bevor Sie diese implementieren.

Sie können Sicherheitsgruppen nur während der Erstellung eines Clusters angeben. Sie können keine Sicherheitsgruppen zu Clustern oder Cluster-Instances hinzufügen, während ein Cluster ausgeführt wird. Sie können jedoch Regeln in vorhandenen Sicherheitsgruppen bearbeiten, hinzufügen und entfernen. Die Regeln treten in Kraft, sobald Sie sie speichern.

Sicherheitsgruppen sind standardmäßig einschränkend. Wenn keine Regel hinzugefügt wird, die Datenverkehr zulässt, wird der Datenverkehr zurückgewiesen. Wenn es mehr als eine Regel für denselben Datenverkehr und dieselbe Quelle gibt, wird die toleranteste Regel angewendet. Wenn es beispielsweise eine Regel gibt, die SSH-Verbindungen von der IP-Adresse 192.0.2.12/32 zulässt, und eine weitere Regel, die dem gesamten TCP-Datenverkehr Zugriff aus dem Bereich 192.0.2.0/24 gewährt, hat die Regel Vorrang, die dem gesamten TCP-Datenverkehr aus dem Bereich Zugriff gewährt, der 192.0.2.12 einschließt. In diesem Fall könnte der Client unter 192.0.2.12 mehr Zugriff erhalten als beabsichtigt.

Important

Seien Sie vorsichtig, wenn Sie Regeln für offene Ports für Sicherheitsgruppen bearbeiten. Stellen Sie sicher, dass Sie Regeln hinzufügen, die nur Datenverkehr von vertrauenswürdigen und authentifizierten Clients für die Protokolle und Ports zulassen, die zum Ausführen Ihrer Workloads erforderlich sind.

Sie können Amazon EMR Block Public Access in jeder Region konfigurieren, die Sie verwenden, um die Cluster-Erstellung zu verhindern, wenn eine Regel den öffentlichen Zugriff auf beliebige Ports zulässt, die Sie nicht einer Liste von Ausnahmen hinzufügen. Für AWS Konten, die nach Juli 2019 erstellt wurden, ist Amazon EMR Block Public Access standardmäßig aktiviert. Für AWS Konten, die vor Juli 2019 einen Cluster erstellt haben, ist Amazon EMR Block Public Access standardmäßig deaktiviert. Weitere Informationen finden Sie unter [Verwenden von Amazon EMR Block Public Access](#).

Themen

- [Arbeiten mit von Amazon EMR verwalteten Sicherheitsgruppen](#)
- [Arbeiten mit zusätzlichen Sicherheitsgruppen für einen Amazon EMR-Cluster](#)
- [Angaben von Amazon-EMR-verwalteten und zusätzlichen Sicherheitsgruppen](#)
- [EC2 Sicherheitsgruppen für EMR Notebooks angeben](#)
- [Verwenden von Amazon EMR Block Public Access](#)

 Note

Amazon EMR ist bestrebt, integrative Alternativen für potenziell anstößige oder nicht inklusive Branchenbegriffe wie „Master“ und „Slave“ zu verwenden. Wir haben auf eine neue Terminologie umgestellt, um ein umfassenderes Erlebnis zu bieten und Ihnen das Verständnis der Servicekomponenten zu erleichtern.

Wir beschreiben „Knoten“ jetzt als Instances und Amazon-EMR-Instance-Typen als Primär, Core, und Aufgaben-Instances. Während der Umstellung finden Sie möglicherweise immer noch ältere Verweise auf die veralteten Begriffe, z. B. solche, die sich auf Sicherheitsgruppen für Amazon EMR beziehen.

Arbeiten mit von Amazon EMR verwalteten Sicherheitsgruppen

 Note

Amazon EMR ist bestrebt, integrative Alternativen für potenziell anstößige oder nicht inklusive Branchenbegriffe wie „Master“ und „Slave“ zu verwenden. Wir haben auf eine neue Terminologie umgestellt, um ein umfassenderes Erlebnis zu bieten und Ihnen das Verständnis der Servicekomponenten zu erleichtern.

Wir beschreiben „Knoten“ jetzt als Instances und Amazon-EMR-Instance-Typen als Primär, Core, und Aufgaben-Instances. Während der Umstellung finden Sie möglicherweise immer noch ältere Verweise auf die veralteten Begriffe, z. B. solche, die sich auf Sicherheitsgruppen für Amazon EMR beziehen.

Mit der Primär-Instance und den Core- und Aufgaben-Instances in einem Cluster sind verschiedene verwaltete Sicherheitsgruppen verknüpft. Sie benötigen eine zusätzliche verwaltete Sicherheitsgruppe für den Servicezugriff, wenn Sie einen Cluster in einem privaten Subnetz erstellen. Weitere Informationen zur Rolle von verwalteten Sicherheitsgruppen in Bezug auf Ihre Netzwerkkonfiguration finden Sie unter [Amazon VPC-Optionen beim Starten eines Clusters](#).

Wenn Sie verwaltete Sicherheitsgruppen für einen Cluster angeben, müssen Sie für alle verwalteten Sicherheitsgruppen denselben Typ von Sicherheitsgruppe (Standard oder benutzerdefiniert) verwenden. Sie können beispielsweise nicht eine benutzerdefinierte Sicherheitsgruppe für die Primär-Instance angeben und dann keine benutzerdefinierte Sicherheitsgruppe für die Core- und Aufgaben-Instances angeben.

Wenn Sie standardmäßige verwaltete Sicherheitsgruppen verwenden, müssen Sie diese beim Erstellen eines Clusters nicht angeben. Amazon EMR verwendet automatisch die Standardeinstellungen. Wenn die Standardeinstellungen in der VPC des Clusters noch nicht vorhanden sind, werden sie außerdem von Amazon EMR erstellt. Amazon EMR erstellt sie auch, wenn Sie sie explizit angeben und sie noch nicht existieren.

Sie können die Regeln in verwalteten Sicherheitsgruppen nach der Erstellung der Cluster bearbeiten. Wenn Sie einen neuen Cluster erstellen, überprüft Amazon EMR die Regeln in den von Ihnen angegebenen verwalteten Sicherheitsgruppen und erstellt dann alle fehlenden eingehenden Regeln, die der neue Cluster zusätzlich zu den Regeln benötigt, die möglicherweise zuvor hinzugefügt wurden. Sofern nicht anders definiert, werden alle Regeln, die für standardmäßig Amazon-EMR-verwaltete Sicherheitsgruppen gelten, auch den von Ihnen angegebenen benutzerdefinierten von Amazon EMR verwalteten Sicherheitsgruppen hinzugefügt.

Die standardmäßigen verwalteten Sicherheitsgruppen sind:

- ElasticMapReduce-primär

Informationen zu den Regeln in dieser Sicherheitsgruppe finden Sie unter [Von Amazon EMR verwaltete Sicherheitsgruppe für die primäre Instance \(öffentliche Subnetze\)](#).

- ElasticMapReduce-Kern

Informationen zu den Regeln in dieser Sicherheitsgruppe finden Sie unter [Von Amazon EMR verwaltete Sicherheitsgruppe für Core- und Aufgaben-Instances \(öffentliche Subnetze\)](#).

- ElasticMapReduce-Primär-Privat

Informationen zu den Regeln in dieser Sicherheitsgruppe finden Sie unter [Von Amazon EMR verwaltete Sicherheitsgruppe für die Master-Instance \(private Subnetze\)](#).

- ElasticMapReduce-Core-Privat

Informationen zu den Regeln in dieser Sicherheitsgruppe finden Sie unter [Von Amazon EMR verwaltete Sicherheitsgruppe für Core- und Aufgaben-Instances \(private Subnetze\)](#).

- ElasticMapReduce-ServiceAccess

Informationen zu den Regeln in dieser Sicherheitsgruppe finden Sie unter [Von Amazon EMR verwaltete Sicherheitsgruppe für den Servicezugriff \(private Subnetze\)](#).

Von Amazon EMR verwaltete Sicherheitsgruppe für die primäre Instance (öffentliche Subnetze)

Die standardmäßig verwaltete Sicherheitsgruppe für die primäre Instance in öffentlichen Subnetzen hat den Gruppennamen `-primary`. ElasticMapReduce Sie hat die folgenden Regeln. Wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe angeben, fügt Amazon EMR Ihrer benutzerdefinierten Sicherheitsgruppe dieselben Regeln hinzu.

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details

Regeln für eingehenden Datenverkehr

Alles ICMP-IPv4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für die Primär-Instance. Mit anderen Worten, dieselbe Sicherheitsgruppe, in der die Regel angezeigt wird.	Diese reflexiven Regeln ermöglichen eingehenden Datenverkehr aus allen mit der angegebenen Sicherheitsgruppe verknüpften Instances. Die Verwendung der standardmäßig verwalteten Sicherheitsgruppe <code>ElasticMapReduce-primary</code> für mehrere Cluster ermöglicht den Core- und Aufgabenknoten dieser Cluster die gegenseitige Kommunikation über ICMP oder einen TCP- oder UDP-Port. Sie geben benutzerdefinierte verwaltete Sicherheitsgruppen an, um den Cluster-übergreifenden Zugriff einzuschränken.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		
Alles ICMP-IPv4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe, die für Core- und Aufgabenknoten angegeben wurde.	Diese Regeln lassen jeden eingehenden ICMP-Datenverkehr und jeden Datenverkehr über TCP- oder UDP-Ports aus allen Core- und Aufgaben-Instances zu, die mit der angegebenen Sicherheitsgruppe verknüpft sind, auch wenn sich die Instances in verschiedenen Clustern befinden.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details
Benutzerdefiniert	TCP	8443	Verschiedene Amazon-IP-Adressbereiche	Diese Regeln ermöglichen dem Cluster-Manager die Kommunikation mit dem Primärknoten.

Um vertrauenswürdigen Quellen SSH-Zugriff auf die primäre Sicherheitsgruppe mit der Konsole zu gewähren

Um Ihre Sicherheitsgruppen zu bearbeiten, benötigen Sie die Berechtigung, Sicherheitsgruppen für die VPC zu verwalten, in der sich der Cluster befindet. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Ändern von Benutzerberechtigungen](#) und unter [Beispielrichtlinie](#), die die Verwaltung von EC2 Sicherheitsgruppen ermöglicht.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie Clusters (Cluster) aus. Wählen Sie die ID des Clusters aus, den Sie ändern möchten.
3. Erweitern Sie im Bereich Netzwerk und Sicherheit die Dropdownliste EC2 Sicherheitsgruppen (Firewall).
4. Wählen Sie unter Primärer Knoten Ihre Sicherheitsgruppe aus.
5. Wählen Sie Edit inbound rules (Regeln für eingehenden Datenverkehr bearbeiten) aus.
6. Suchen Sie mit den folgenden Einstellungen nach einer Regel für eingehenden Datenverkehr, die öffentlichen Zugriff ermöglicht. Falls sie existiert, wählen Sie Löschen, um sie zu entfernen.

- Typ

SSH

- Port

22

- Quelle

Benutzerdefiniert 0.0.0.0/0

 Warning

Vor Dezember 2020 gab es eine vorkonfigurierte Regel, die eingehenden Datenverkehr auf Port 22 aus allen Quellen zuließ. Diese Regel wurde erstellt, um die ersten SSH-Verbindungen zum Primärknoten zu vereinfachen. Wir empfehlen Ihnen dringend, diese Eingangsregel zu entfernen und den Datenverkehr auf vertrauenswürdige Quellen zu beschränken.

7. Scrollen Sie zum Ende der Regelliste und wählen Sie Regel hinzufügen.
8. Wählen Sie für Type (Typ) SSH aus.

Wenn Sie SSH auswählen, wird automatisch TCP für Protokoll und 22 für Portbereich eingegeben.

9. Wählen Sie als Quelle Meine IP aus, um Ihre IP-Adresse automatisch als Quelladresse hinzuzufügen. Sie können auch einen Bereich benutzerdefinierter vertrauenswürdiger Client-IP-Adressen hinzufügen oder zusätzliche Regeln für andere Clients erstellen. In vielen Netzwerkumgebungen werden IP-Adressen dynamisch zugewiesen, sodass Sie in Zukunft möglicherweise Ihre IP-Adressen für vertrauenswürdige Clients aktualisieren müssen.
10. Wählen Sie Speichern.
11. Wählen Sie optional im Bereich Netzwerk und Sicherheit unter Kern- und Taskknoten die andere Sicherheitsgruppe aus und wiederholen Sie die obigen Schritte, um dem SSH-Client den Zugriff auf Core- und Taskknoten zu ermöglichen.

Von Amazon EMR verwaltete Sicherheitsgruppe für Core- und Aufgaben-Instances (öffentliche Subnetze)

Die standardmäßig verwaltete Sicherheitsgruppe für Core- und Task-Instances in öffentlichen Subnetzen hat den Gruppennamen `-core`. ElasticMapReduce Für die verwaltete Standardsicherheitsgruppe gelten die folgenden Regeln. Amazon EMR fügt die gleichen Regeln hinzu, wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe angeben.

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details

Regeln für eingehenden Datenverkehr

Alle ICMP-IPV4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für Core- und Aufgaben-Instances. Mit anderen Worten, dieselbe Sicherheitsgruppe, in der die Regel angezeigt wird.	Diese reflexiven Regeln ermöglichen eingehenden Datenverkehr aus allen mit der angegebenen Sicherheitsgruppe verknüpften Instances. Die Verwendung der standardmäßig verwalteten Sicherheitsgruppe ElasticMapReduce-core für mehrere Cluster ermöglicht den Core- und Aufgaben-Instances dieser Cluster die gegenseitige Kommunikation über ICMP oder einen TCP- oder UDP-Port. Sie geben benutzerdefinierte verwaltete Sicherheitsgruppen an, um den Cluster-übergreifenden Zugriff einzuschränken.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		
Alles ICMP-IPV4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für die Primär-Instance.	Diese Regeln lassen jeden eingehenden ICMP-Datenverkehr und jeden Datenverkehr über TCP- oder UDP-Ports aus allen Primär-Instances zu, die mit der angegebenen Sicherheitsgruppe verknüpft sind, auch wenn sich die Instances in verschiedenen Clustern befinden.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		

Von Amazon EMR verwaltete Sicherheitsgruppe für die Master-Instance (private Subnetze)

Die standardmäßig verwaltete Sicherheitsgruppe für die primäre Instanz in privaten Subnetzen hat den Gruppennamen `-Primary-Private`. ElasticMapReduce Für die verwaltete Standardsicherheitsgruppe gelten die folgenden Regeln. Amazon EMR fügt die gleichen Regeln hinzu, wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe angeben.

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details

Regeln für eingehenden Datenverkehr

Alle ICMP-IPv4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für die Primär-Instance. Mit anderen Worten, dieselbe Sicherheitsgruppe, in der die Regel angezeigt wird.	Diese reflexiven Regeln ermöglichen eingehenden Datenverkehr aus allen mit der angegebenen Sicherheitsgruppe verknüpften Instances, die aus dem privaten Subnetz erreichbar sind. Die Verwendung der standardmäßigen verwalteten Sicherheitsgruppe ElasticMapReduce-Primary-Private für mehrere Cluster ermöglicht den Core- und Aufgabenknoten dieser Cluster die gegenseitige Kommunikation über ICMP oder einen TCP- oder UDP-Port. Sie geben benutzerdefinierte verwaltete Sicherheitsgruppen an, um den Cluster-übergreifenden Zugriff einzuschränken.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		
Alles ICMP-IPV4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für Core- und Aufgabenknoten.	Diese Regeln lassen jeden eingehenden ICMP-Datenverkehr und jeden Datenverkehr über TCP- oder UDP-Ports aus allen Core- und Aufgaben-Instances zu, die mit der angegebenen Sicherheitsgruppe verknüpft und aus dem privaten Subnetz erreichbar sind, auch wenn sich die Instances in verschiedenen Clustern befinden.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		
HTTPS (8443)	TCP	8443	Die Gruppen-ID der verwalteten Sicherheitsgruppe für den Servicezugriff in einem privaten Subnetz.	Diese Regel ermöglicht dem Cluster-Manager die Kommunikation mit dem Primärknoten.

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details

Regeln für ausgehenden Datenverkehr

Gesamter Datenverkehr	Alle	Alle	0.0.0.0/0	Stellt ausgehenden Zugriff auf das Internet zu.
Custom TCP	TCP	9443	Die Gruppen-ID der verwalteten Sicherheitsgruppe für den Servicezugriff in einem privaten Subnetz.	Wenn die obige Standardregel „Gesamter Datenverkehr“ für ausgehenden Datenverkehr entfernt wird, ist diese Regel eine Mindestanforderung für Amazon EMR 5.30.0 und höher.  Note Amazon EMR fügt diese Regel nicht hinzu, wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe verwenden.
Custom TCP	TCP	80 (http) oder 443 (https)	Die Gruppen-ID der verwalteten Sicherheitsgruppe für den Servicezugriff in einem privaten Subnetz.	Wenn die obige Standardregel „Gesamter Datenverkehr“ für ausgehenden Datenverkehr entfernt wird, ist diese Regel eine Mindestanforderung für Amazon EMR 5.30.0 und höher, um über https eine Verbindung zu Amazon S3 herzustellen.  Note Amazon EMR fügt diese Regel nicht hinzu, wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe verwenden.

Von Amazon EMR verwaltete Sicherheitsgruppe für Core- und Aufgaben-Instances (private Subnetze)

Die standardmäßig verwaltete Sicherheitsgruppe für Core- und Task-Instances in privaten Subnetzen hat den Gruppennamen `-Core-Private`. ElasticMapReduce Für die verwaltete Standardsicherheitsgruppe gelten die folgenden Regeln. Amazon EMR fügt die gleichen Regeln hinzu, wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe angeben.

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details

Regeln für eingehenden Datenverkehr

Alle ICMP-IPV4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für Core- und Aufgaben-Instances. Mit anderen Worten, dieselbe Sicherheitsgruppe, in der die Regel angezeigt wird.	Diese reflexiven Regeln ermöglichen eingehenden Datenverkehr aus allen mit der angegebenen Sicherheitsgruppe verknüpften Instances. Die Verwendung der standardmäßig verwalteten Sicherheitsgruppe <code>ElasticMapReduce-core</code> für mehrere Cluster ermöglicht den Core- und Aufgaben-Instances dieser Cluster die gegenseitige Kommunikation über ICMP oder einen TCP- oder UDP-Port. Sie geben benutzerdefinierte verwaltete Sicherheitsgruppen an, um den Cluster-übergreifenden Zugriff einzuschränken.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		
Alles ICMP-IPV4	Alle	N/A	Die Gruppen-ID der verwalteten Sicherheitsgruppe für die Primär-Instance.	Diese Regeln lassen jeden eingehenden ICMP-Datenverkehr und jeden Datenverkehr über TCP- oder UDP-Ports aus allen Primär-Instances zu, die mit der angegebenen Sicherheitsgruppe verknüpft sind, auch wenn sich die Instances in verschiedenen Clustern befinden.
Alle TCP	TCP	Alle		
Alle UDP	UDP	Alle		

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details
HTTPS (8443)	TCP	8443	Die Gruppen-ID der verwalteten Sicherheitsgruppe für den Servicezugriff in einem privaten Subnetz.	Diese Regel ermöglicht dem Cluster-Manager die Kommunikation mit Core- und Aufgabenknoten.

Regeln für ausgehenden Datenverkehr

Gesamter Datenverkehr	Alle	Alle	0.0.0.0/0	Weitere Informationen finden Sie unter Regeln für ausgehenden Datenverkehr bearbeiten weiter unten in diesem Dokument.
Custom TCP	TCP	80 (http) oder 443 (https)	Die Gruppen-ID der verwalteten Sicherheitsgruppe für den Servicezugriff in einem privaten Subnetz.	Wenn die obige Standardregel „Gesamter Datenverkehr“ für ausgehenden Datenverkehr entfernt wird, ist diese Regel eine Mindestanforderung für Amazon EMR 5.30.0 und höher, um über https eine Verbindung zu Amazon S3 herzustellen.

 **Note**

Amazon EMR fügt diese Regel nicht hinzu, wenn Sie eine benutzerdefinierte verwaltete Sicherheitsgruppe verwenden.

Regeln für ausgehenden Datenverkehr bearbeiten

Standardmäßig erstellt Amazon EMR diese Sicherheitsgruppe mit ausgehenden Regeln, die den gesamten ausgehenden Datenverkehr auf allen Protokollen und Ports zulassen. Das Zulassen des gesamten ausgehenden Datenverkehrs ist ausgewählt, da für verschiedene Amazon-EMR- und Kundenanwendungen, die auf Amazon-EMR-Clustern ausgeführt werden können,

möglicherweise unterschiedliche Ausgangsregeln erforderlich sind. Amazon EMR kann diese spezifischen Einstellungen bei der Erstellung von Standardsicherheitsgruppen nicht antizipieren. Sie können den ausgehenden Datenverkehr in Ihren Sicherheitsgruppen einschränken, sodass nur die Regeln berücksichtigt werden, die Ihren Anwendungsfällen und Sicherheitsrichtlinien entsprechen. Für diese Sicherheitsgruppe sind mindestens die folgenden Regeln für ausgehenden Datenverkehr erforderlich, für einige Anwendungen sind jedoch möglicherweise zusätzliche Regeln für ausgehenden Datenverkehr erforderlich.

Typ	Protokoll (Protokoll)	Port-Bereich	Bestimmungsort	Details
Alle TCP	TCP	Alle	pl- xxxxxxxx	Verwaltete Präfixliste von Amazon S3 com.amazonaws. MyRegion .s3.
Gesamter Datenverkehr	Alle	Alle	sg- xxxxxxxx xxxxxxxx	Die ID der Sicherheitsgruppe ElasticMapReduce-Core-Private .
Gesamter Datenverkehr	Alle	Alle	sg- xxxxxxxx xxxxxxxx	Die ID der Sicherheitsgruppe ElasticMapReduce-Primary-Private .
Custom TCP	TCP	9443	sg- xxxxxxxx xxxxxxxx	Die ID der Sicherheitsgruppe ElasticMapReduce-ServiceAccess .

Von Amazon EMR verwaltete Sicherheitsgruppe für den Servicezugriff (private Subnetze)

Die standardmäßig verwaltete Sicherheitsgruppe für den Dienstzugriff in privaten Subnetzen hat den Gruppennamen ElasticMapReduce - ServiceAccess. Sie besitzt Regeln für den eingehenden und den ausgehenden Datenverkehr, die Datenverkehr über HTTPS (Port 8443, Port 9443) mit den anderen verwalteten Sicherheitsgruppen in privaten Subnetzen zulassen. Diese Regeln ermöglichen dem Cluster-Manager die Kommunikation mit dem Primärknoten und mit Kern- und Aufgabenknoten. Dieselben Regeln sind erforderlich, wenn Sie benutzerdefinierte Sicherheitsgruppen verwenden.

Typ	Protokoll (Protokoll)	Port-Bereich	Quelle	Details

Regeln für eingehenden Datenverkehr Erforderlich für EMR-Cluster mit Amazon EMR ab Version 5.30.0.

Custom TCP	TCP	9443	Die Gruppen-ID der verwalteten Sicherheitsgruppe für die Primär-Instance.	Diese Regel ermöglicht die Kommunikation zwischen der Sicherheitsgruppe der Primär-Instance und der Sicherheitsgruppe des Servicezugriffs.

Regeln für ausgehenden Datenverkehr erforderlich für alle Amazon-EMR-Cluster

Custom TCP	TCP	8443	Die Gruppen-ID der verwalteten Sicherheitsgruppe für die Primär-Instance.	Diese Regeln ermöglichen dem Cluster-Manager die Kommunikation mit dem Primärknoten und mit Kern- und Aufgabenknoten.
Custom TCP	TCP	8443	Die Gruppen-ID der verwalteten Sicherheitsgruppe für Core- und Aufgaben-Instances.	Diese Regeln ermöglichen dem Cluster-Manager die Kommunikation mit dem Primärknoten und mit Kern- und Aufgabenknoten.

Arbeiten mit zusätzlichen Sicherheitsgruppen für einen Amazon EMR-Cluster

Sie können zusätzliche Sicherheitsgruppen unabhängig davon verwenden, ob Sie die standardmäßigen verwalteten Sicherheitsgruppen verwenden oder benutzerdefinierte verwaltete Sicherheitsgruppen angeben. Mit zusätzlichen Sicherheitsgruppen können Sie den Zugriff auf die einzelnen Cluster und von externen Clients, Ressourcen und Anwendungen anpassen.

Betrachten Sie beispielsweise das folgende Szenario. Es gibt mehrere Cluster, die miteinander kommunizieren müssen. Sie möchten jedoch nur einem bestimmten Teilsatz von Clustern eingehenden SSH-Zugriff auf die Primär-Instance gewähren. Hierzu können Sie für die Cluster den gleichen Satz von verwalteten Sicherheitsgruppen verwenden. Anschließend erstellen Sie zusätzliche Sicherheitsgruppen, die eingehenden SSH-Zugriff von vertrauenswürdigen Clients zulassen, und geben die zusätzlichen Sicherheitsgruppen für die Primär-Instance für jeden Cluster in der Untergruppe an.

Sie können bis zu 15 zusätzliche Sicherheitsgruppen für die primäre Instance, 15 für Core- und Task-Instances und 15 für den Servicezugriff (in privaten Subnetzen) anwenden. Wenn notwendig, können Sie dieselben zusätzlichen Sicherheitsgruppen für Primär-Instances, Core- und Aufgaben-Instances und den Servicezugriff angeben. Die maximale Anzahl von Sicherheitsgruppen und -regeln in Ihrem Konto unterliegt Kontolimits. Weitere Informationen finden Sie unter [Sicherheitsgruppenlimits](#) im Amazon-VPC-Benutzerhandbuch.

Angeben von Amazon-EMR-verwalteten und zusätzlichen Sicherheitsgruppen

Sie können Sicherheitsgruppen mithilfe der AWS Management Console, der AWS CLI, der oder der Amazon EMR-API angeben. Wenn Sie keine Sicherheitsgruppen angeben, erstellt Amazon EMR Standardsicherheitsgruppen. Die Angabe zusätzlicher Sicherheitsgruppen ist optional. Sie können Primär-Instances, Core- und Aufgaben-Instances und dem Servicezugriff (nur private Subnetze) zusätzliche Sicherheitsgruppen zuweisen.

Console

Um Sicherheitsgruppen mit der Konsole anzugeben

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Netzwerk den Pfeil neben EC2 Sicherheitsgruppen (Firewall) aus, um diesen Abschnitt zu erweitern. Unter Primärknoten und Kern- und Aufgabenknoten sind standardmäßig die von Amazon EMR verwalteten Standardsicherheitsgruppen ausgewählt. Wenn Sie ein privates Subnetz verwenden, haben Sie auch die Möglichkeit, eine Sicherheitsgruppe für den Servicezugriff auszuwählen.

4. Um Ihre von Amazon EMR verwaltete Sicherheitsgruppe zu ändern, verwenden Sie das Dropdownmenü Sicherheitsgruppen auswählen, um eine andere Option aus der Optionsliste der Von Amazon EMR verwalteten Sicherheitsgruppen auszuwählen. Sie haben eine von Amazon EMR verwaltete Sicherheitsgruppe sowohl für den Primärknoten als auch für den Core- und Aufgabenknoten.
5. Um benutzerdefinierte Sicherheitsgruppen hinzuzufügen, verwenden Sie dasselbe Dropdownmenü Sicherheitsgruppen auswählen, um bis zu vier benutzerdefinierte Sicherheitsgruppen aus der Optionsliste Benutzerdefinierte Sicherheitsgruppen auszuwählen. Sie können bis zu vier benutzerdefinierte Sicherheitsgruppen sowohl für den Primärknoten als auch für den Kern- und Aufgabenknoten einrichten.
6. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
7. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

Angeben von Sicherheitsgruppen mit der AWS CLI

Um Sicherheitsgruppen mithilfe von anzugeben, verwenden AWS CLI Sie den `create-cluster` Befehl mit den folgenden `--ec2-attributes` Optionsparametern:

Parameter	Beschreibung
<code>EmrManagedPrimarySecurityGroup</code>	Verwenden Sie diesen Parameter, um eine benutzerdefinierte verwaltete Sicherheitsgruppe für die Primär-Instance anzugeben. Wenn dieser Parameter angegeben wird, muss auch <code>EmrManagedCoreSecurityGroup</code> angegeben werden. Für Cluster in privaten Subnetzen muss auch <code>ServiceAccessSecurityGroup</code> angegeben werden.
<code>EmrManagedCoreSecurityGroup</code>	Verwenden Sie diesen Parameter, um eine benutzerdefinierte verwaltete Sicherheitsgruppe für Core- und Aufgaben-Instances anzugeben. Wenn dieser Parameter angegeben wird, muss auch <code>EmrManagedPrimarySecurityGroup</code> angegeben werden. Für Cluster in

Parameter	Beschreibung
	privaten Subnetzen muss auch <code>ServiceAccessSecurityGroup</code> angegeben werden.
<code>ServiceAccessSecurityGroup</code>	Verwenden Sie diesen Parameter, um eine benutzerdefinierte verwaltete Sicherheitsgruppe für den Servicezugriff anzugeben. Dies gilt nur für Cluster in privaten Subnetzen. Die Sicherheitsgruppe, als die Sie angeben, <code>ServiceAccessSecurityGroup</code> sollte nicht für andere Zwecke verwendet werden und sollte auch für Amazon EMR reserviert werden. Wenn dieser Parameter angegeben wird, muss auch <code>EmrManagedPrimarySecurityGroup</code> angegeben werden.
<code>AdditionalPrimarySecurityGroups</code>	Verwenden Sie diesen Parameter, um bis zu vier zusätzliche verwaltete Sicherheitsgruppen für die Primär-Instance anzugeben.
<code>AdditionalCoreSecurityGroups</code>	Verwenden Sie diesen Parameter, um bis zu vier zusätzliche verwaltete Sicherheitsgruppen für die Core- und Aufgaben-Instances anzugeben.

Example – spezifizieren Sie benutzerdefinierte, von Amazon EMR verwaltete Sicherheitsgruppen und zusätzliche Sicherheitsgruppen

Im folgenden Beispiel werden benutzerdefinierte Amazon-EMR-verwaltete Sicherheitsgruppen für einen Cluster in einem privaten Subnetz, mehrere zusätzliche Sicherheitsgruppen für die Master-Instance und eine einzelne zusätzliche Sicherheitsgruppe für Core- und Aufgaben-Instances angegeben.

Note

Linux-Zeilenumbruchzeichen (\) sind aus Gründen der Lesbarkeit enthalten. Sie können entfernt oder in Linux-Befehlen verwendet werden. Entfernen Sie sie unter Windows oder ersetzen Sie sie durch ein Caret-Zeichen (^).

```
aws emr create-cluster --name "ClusterCustomManagedAndAdditionalSGs" \
--release-label emr-emr-7.9.0 --applications Name=Hue Name=Hive \
Name=Pig --use-default-roles --ec2-attributes \
SubnetIds=subnet-xxxxxxxxxxxx,KeyName=myKey,\
ServiceAccessSecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedPrimarySecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedCoreSecurityGroup=sg-xxxxxxxxxxxx,\
AdditionalPrimarySecurityGroups=['sg-xxxxxxxxxxxx',\
'sg-xxxxxxxxxxxx', 'sg-xxxxxxxxxxxx'],\
AdditionalCoreSecurityGroups=sg-xxxxxxxxxxxx \
--instance-type m5.xlarge
```

Weitere Informationen finden Sie unter [create-cluster](#) in der AWS CLI -Befehlsreferenz.

EC2 Sicherheitsgruppen für EMR Notebooks angeben

Wenn Sie ein EMR Notebook erstellen, wird der Netzwerkdatenverkehr zwischen dem EMR Notebook und dem Amazon-EMR-Cluster mithilfe von zwei Sicherheitsgruppen gesteuert, wenn Notebook-verwendet wird. Die Standard-Sicherheitsgruppen verfügen über Mindestregeln, die nur Netzwerkdatenverkehr zwischen dem EMR-Notebooks-Service und den Clustern zulassen, an die die Notebooks angefügt sind.

Ein EMR-Notebook verwendet [Apache Livy](#), um über einen Proxy über den TCP-Port 18888 mit dem Cluster zu kommunizieren. Indem Sie benutzerdefinierte Sicherheitsgruppen mit an Ihre Umgebung angepassten Regeln erstellen, können Sie den Netzwerkdatenverkehr so einschränken, dass nur ein Teil der Notebooks Code innerhalb des Notebook-Editors auf bestimmten Clustern ausführen kann. Der Cluster verwendet Ihre benutzerdefinierte Sicherheit zusätzlich zu den Standardsicherheitsgruppen für den Cluster. Weitere Informationen finden Sie unter [Steuerung des Netzwerkverkehrs mit Sicherheitsgruppen](#) im Verwaltungshandbuch für Amazon EMR und [EC2 Sicherheitsgruppen für EMR Notebooks angeben](#).

EC2 Standard-Sicherheitsgruppe für die primäre Instance

Die EC2 Standardsicherheitsgruppe für die primäre Instance ist zusätzlich zu den Sicherheitsgruppen des Clusters für die primäre Instance mit der primären Instance verknüpft.

Gruppenname: ElasticMapReduceEditors-Livy

Regeln

- Eingehend

Erlauben Sie den TCP-Port 18888 von allen Ressourcen in der EC2 Standardsicherheitsgruppe für EMR Notebooks

- Ausgehend

Keine

EC2Standardsicherheitsgruppe für EMR Notebooks

Die EC2 Standardsicherheitsgruppe für das EMR-Notizbuch ist dem Notebook-Editor für jedes EMR-Notizbuch zugeordnet, dem es zugewiesen ist.

Gruppenname: -Editor ElasticMapReduceEditors

Regeln

- Eingehend

Keine

- Ausgehend

Erlauben Sie TCP-Port 18888 für alle Ressourcen in der EC2 Standardsicherheitsgruppe für EMR Notebooks.

Benutzerdefinierte EC2 Sicherheitsgruppe für EMR Notebooks bei der Verknüpfung von Notebooks mit Git-Repositorys

Um ein Git-Repository mit Ihrem Notebook verknüpfen zu können, muss die Sicherheitsgruppe für das EMR Notebook eine Regel für ausgehenden Datenverkehr enthalten, damit das Notebook

Datenverkehr an das Internet weiterleiten kann. Es wird empfohlen, zu diesem Zweck eine neue Sicherheitsgruppe zu erstellen. Beim Aktualisieren der Standardsicherheitsgruppe ElasticMapReduceEditors-Editor gelten möglicherweise dieselben Regeln für ausgehende Nachrichten auch für andere Notebooks, die zu dieser Sicherheitsgruppe gehören.

Regeln

- Eingehend

Keine

- Ausgehend

Erlauben Sie dem Notebook, Datenverkehr über den Cluster an das Internet zu leiten, wie im folgenden Beispiel veranschaulicht. Der Wert 0.0.0.0/0 wird für Beispielszwecke verwendet. Sie können diese Regel ändern, um die IP-Adressen für Ihre Git-basierten Repositorys anzugeben.

Typ	Protocol (Protokoll)	Port-Bereich	Bestimmungsort
Benutzerdefinierte TCP-Regel	TCP	18888	SG-
HTTPS	TCP	443	0.0.0.0/0

Verwenden von Amazon EMR Block Public Access

Amazon EMR Block Public Access (BPA) verhindert, dass Sie einen Cluster in einem öffentlichen Subnetz starten, wenn der Cluster über eine Sicherheitskonfiguration verfügt, die eingehenden Datenverkehr von öffentlichen IP-Adressen an einem Port zulässt.

Important

Den öffentlichen Zugriff blockieren ist standardmäßig aktiviert. Um den Kontoschutz zu erhöhen, empfehlen wir, ihn aktiviert zu lassen.

Grundlegendes zum Blockieren des öffentlichen Zugriffs

Sie können die Konfiguration Block Public Access auf Kontoebene verwenden, um den öffentlichen Netzwerkzugriff auf Amazon-EMR-Cluster zentral zu verwalten.

Wenn ein Benutzer von Ihnen einen Cluster AWS-Konto startet, überprüft Amazon EMR die Portregeln in der Sicherheitsgruppe für den Cluster und vergleicht sie mit Ihren Regeln für eingehenden Datenverkehr. Wenn die Sicherheitsgruppe über eine Regel für eingehenden Datenverkehr verfügt, die Ports zu den öffentlichen IP-Adressen IPv4 0.0.0.0/0 oder IPv6 :::/0 öffnet, und diese Ports nicht als Ausnahmen für Ihr Konto angegeben sind, lässt Amazon EMR den Benutzer den Cluster nicht erstellen.

Wenn ein Benutzer die Sicherheitsgruppenregeln für einen laufenden Cluster in einem öffentlichen Subnetz so ändert, dass er über eine Regel für den öffentlichen Zugriff verfügt, die gegen die BPA-Konfiguration für Ihr Konto verstößt, widerruft Amazon EMR die neue Regel, sofern es dazu berechtigt ist. Wenn Amazon EMR nicht berechtigt ist, die Regel zu widerrufen, wird im AWS Health -Dashboard ein Ereignis erstellt, das den Verstoß beschreibt. Informationen dazu, wie Sie Amazon EMR die Berechtigung zum Widerrufen der Regel erteilen, finden Sie unter [Amazon EMR so konfigurieren, dass Sicherheitsgruppenregeln aufgehoben werden](#).

Den öffentlichen Zugriff blockieren ist standardmäßig für alle Cluster in jedem AWS-Region für Ihr AWS-Konto aktiviert. BPA gilt für den gesamten Lebenszyklus eines Clusters, gilt jedoch nicht für Cluster, die Sie in privaten Subnetzen erstellen. Sie können Ausnahmen von der BPA-Regel konfigurieren. Port 22 ist standardmäßig eine Ausnahme. Weitere Informationen zur Einstellung finden Sie unter [Konfigurieren von Block Public Access](#).

Konfigurieren von Block Public Access

Sie können die Sicherheitsgruppen und die Konfiguration zum Sperren des öffentlichen Zugriffs in Ihren Konten jederzeit aktualisieren.

Sie können die Einstellungen für den öffentlichen Zugriff (Block Public Access, BPA) mit der AWS Management Console, der AWS Command Line Interface (AWS CLI) und der Amazon EMR-API ein- und ausschalten. Die Einstellungen gelten für Ihr gesamtes Konto auf einer Region-by-Region bestimmten Basis. Um die Clustersicherheit aufrechtzuerhalten, wird die Verwendung von BPA empfohlen.

Console

Um den öffentlichen Zugriff blockieren mit der Konsole zu konfigurieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie dann die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie in der oberen Navigationsleiste die Region aus, die Sie konfigurieren möchten, sofern sie nicht bereits ausgewählt ist.
3. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Öffentlichen Zugriff blockieren aus.
4. Führen Sie unter Block public access settings (Einstellungen für die Sperrung des öffentlichen Zugriffs) die folgenden Schritte aus.

Zu ...	Vorgehensweise
Block Public Access aktivieren oder deaktivieren	Wählen Sie Bearbeiten, wählen Sie je nach Bedarf Einschalten oder Ausschalten und wählen Sie dann Speichern.
Ports in der Liste der Ausnahmen bearbeiten	1. Wählen Sie Bearbeiten und suchen Sie den Abschnitt Ausnahmen für den Portbereich. 2. Um der Liste der Ausnahmen Ports hinzuzufügen, wählen Sie Add a port range (Port-Bereich hinzufügen) aus und geben Sie einen neuen Port oder Port-Bereich ein. Wiederholen Sie den Vorgang für jeden Port oder Port-Bereich, der hinzugefügt werden soll. 3. Um einen Port oder Portbereich zu entfernen, wählen Sie das Entfernen neben dem Eintrag in der Liste Portbereiche.

Zu ...	Vorgehensweise
	4. Wählen Sie Speichern.

AWS CLI

Um den öffentlichen Zugriff blockieren zu konfigurieren, verwenden Sie den AWS CLI

- Verwenden Sie den `aws emr put-block-public-access-configuration`-Befehl, um Block Public Access zu konfigurieren, wie in den folgenden Beispielen gezeigt.

Zu ...	Vorgehensweise
Block Public Access aktivieren	Legen Sie <code>BlockPublicSecurityGroupRules</code> wie im folgenden Beispiel gezeigt auf <code>true</code> fest. Damit der Cluster gestartet werden kann, darf keine Sicherheitsgruppe, die einem Cluster zugeordnet ist, über eine Regel für eingehenden Datenverkehr verfügen, die den öffentlichen Zugriff zulässt. <pre>aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=true</pre>

Zu ...	Vorgehensweise
Block Public Access deaktivieren	Legen Sie <code>BlockPublicSecurityGroupRules</code> wie im folgenden Beispiel gezeigt auf <code>false</code> fest. Sicherheitsgruppen, die einem Cluster zugeordnet sind, können Regeln für eingehenden Datenverkehr aufweisen, die öffentlichen Zugriff auf beliebige Ports zulassen. Wir empfehlen diese Konfiguration nicht. <pre>aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=false</pre>
Block Public Access aktivieren und Ports als Ausnahmen angeben	Im folgenden Beispiel wird Block Public Access aktiviert und Port 22 sowie die Ports 100-101 werden als Ausnahmen angegeben. Auf diese Weise können Cluster erstellt werden, wenn eine zugeordnete Sicherheitsgruppe über eine Regel für eingehenden Datenverkehr verfügt, die öffentlichen Zugriff auf die Ports 22, 100 oder 101 zulässt. <pre>aws emr put-block-public-access-configuration --block-public-access-configuration '{ "BlockPublicSecurityGroupRules": true, "PermittedPublicSecurityGroupRuleRanges": [{ "MinRange": 22, "MaxRange": 22 }, { "MinRange": 100, "MaxRange": 101 }] }'</pre>

Amazon EMR so konfigurieren, dass Sicherheitsgruppenregeln aufgehoben werden

Amazon EMR benötigt die Erlaubnis, Sicherheitsgruppenregeln zu widerrufen und Ihre Konfiguration für die Blockierung des öffentlichen Zugriffs einzuhalten. Sie können einen der folgenden Ansätze verwenden, um Amazon EMR die erforderliche Genehmigung zu erteilen:

- (Empfohlen) Fügen Sie der Servicerolle die `AmazonEMRServicePolicy_v2`-verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Servicerolle für Amazon EMR \(EMR-Rolle\)](#).
- Erstellen Sie eine neue Inline-Richtlinie, die die `ec2:RevokeSecurityGroupIngress`-Aktion für Sicherheitsgruppen ermöglicht. Weitere Informationen zum Ändern einer Rollenberechtigungsrichtlinie finden Sie unter [Ändern einer Rollenberechtigungsrichtlinie mit der IAM-Konsole](#), der [AWS -API](#) und [AWS CLI](#) im IAM-Benutzerhandbuch.

Beheben von Verletzungen des Blockieren des öffentlichen Zugriffs

Wenn ein Verstoß gegen die Sperrung des öffentlichen Zugriffs auftritt, können Sie ihn mit einer der folgenden Maßnahmen beheben:

- Wenn Sie auf eine Webschnittstelle Ihres Clusters zugreifen möchten, verwenden Sie eine der unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#) beschriebenen Optionen, um über SSH (Port 22) auf die Schnittstelle zuzugreifen.
- Um den Datenverkehr zum Cluster von bestimmten IP-Adressen statt von der öffentlichen IP-Adresse aus zuzulassen, fügen Sie eine Sicherheitsgruppenregel hinzu. Weitere Informationen finden Sie unter [Regeln zu einer Sicherheitsgruppe hinzufügen](#) im Amazon-Handbuch „EC2 Erste Schritte“.
- (Nicht empfohlen) Sie können Amazon-EMR-BPA-Ausnahmen so konfigurieren, dass sie den gewünschten Port oder Portbereich enthalten. Wenn Sie eine BPA-Ausnahme angeben, gehen Sie mit einem ungeschützten Port ein Risiko ein. Wenn Sie beabsichtigen, eine Ausnahme anzugeben, sollten Sie die Ausnahme entfernen, sobald sie nicht mehr benötigt wird. Weitere Informationen finden Sie unter [Konfigurieren von Block Public Access](#).

Identifizieren Sie Cluster, die Sicherheitsgruppenregeln zugeordnet sind

Möglicherweise müssen Sie alle Cluster identifizieren, die einer bestimmten Sicherheitsgruppenregel zugeordnet sind, oder die Sicherheitsgruppenregel für einen bestimmten Cluster finden.

- Wenn Sie die Sicherheitsgruppe kennen, können Sie die zugehörigen Cluster identifizieren, wenn Sie die Netzwerkschnittstellen für die Sicherheitsgruppe finden. Weitere Informationen finden Sie unter [Wie finde ich die Ressourcen, die einer EC2 Amazon-Sicherheitsgruppe zugeordnet sind?](#) in AWS re:Post. Die EC2 Amazon-Instances, die an diese Netzwerkschnittstellen angeschlossen sind, werden mit der ID des Clusters gekennzeichnet, zu dem sie gehören.
- Wenn Sie die Sicherheitsgruppen für einen bekannten Cluster suchen möchten, folgen Sie den Schritten unter [Status und Details des Amazon EMR-Clusters anzeigen](#). Sie finden die Sicherheitsgruppen für den Cluster im Bereich Netzwerk und Sicherheit in der Konsole oder im Ec2InstanceAttributes-Feld unter AWS CLI.

Compliance-Validierung für Amazon EMR

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon EMR im Rahmen mehrerer AWS Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter [AWS Services im Umfang der einzelnen Compliance-Programme](#). Allgemeine Informationen finden Sie unter [AWS -Compliance-Programme](#).

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen in AWS Artifact](#).

Ihre Compliance-Verantwortung bei Verwendung von Amazon EMR hängt von der Vertraulichkeit der Daten, den Compliance-Zielen des Unternehmens und den geltenden Gesetzen und Vorschriften ab. Falls Ihre Nutzung von Amazon EMR der Einhaltung von Standards wie HIPAA, PCI oder FedRAMP unterliegt, bietet Ihnen folgende Ressourcen: AWS

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben. AWS
- Whitepaper „[Architecting for HIPAA Security and Compliance](#)“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen entwickeln können. AWS
- [AWS Ressourcen zur Einhaltung](#) von Vorschriften — Diese Sammlung von Arbeitsmappen und Leitfäden könnte für Ihre Branche und Ihren Standort gelten.
- [AWS Config](#) — Dieser AWS Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- [AWS Security Hub](#)— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus und hilft Ihnen AWS, die Einhaltung der Sicherheitsstandards und bewährten Verfahren der Sicherheitsbranche zu überprüfen.

Ausfallsicherheit bei Amazon EMR

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, 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 Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zur AWS globalen Infrastruktur bietet Amazon EMR mehrere Funktionen, die Sie bei der Unterstützung Ihrer Datenausfallsicherheit und Ihrer Backup-Anforderungen unterstützen.

- Integration in Amazon S3 über EMRFS
- Support für mehrere Master-Knoten

Infrastruktursicherheit in Amazon EMR

Als verwalteter Service ist Amazon EMR durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon EMR zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.

- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Themen

- [Herstellen einer Verbindung mit Amazon EMR über einen Schnittstellen-VPC-Endpunkt](#)

Herstellen einer Verbindung mit Amazon EMR über einen Schnittstellen-VPC-Endpunkt

Sie können eine direkte Verbindung zu Amazon EMR herstellen, indem Sie einen [VPC-Endpunkt \(AWS PrivateLink\)](#) in Ihrer Virtual Private Cloud (VPC) verwenden, anstatt eine Verbindung über das Internet herzustellen. Wenn Sie einen VPC-Endpunkt mit Schnittstelle verwenden, erfolgt die Kommunikation zwischen Ihrer VPC und Amazon EMR vollständig innerhalb des Netzwerks. AWS Jeder VPC-Endpunkt wird durch eine oder mehrere [Elastic Network-Schnittstellen](#) (ENIs) mit privaten IP-Adressen in Ihren VPC-Subnetzen repräsentiert.

Der VPC-Endpunkt der Schnittstelle verbindet Ihre VPC ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder Verbindung direkt mit Amazon EMR. AWS Direct Connect Die Instances in Ihrer VPC benötigen für die Kommunikation mit der API von Amazon EMR keine öffentlichen IP-Adressen.

Um Amazon EMR über Ihre VPC zu verwenden, müssen Sie für die Verbindung eine Instance innerhalb Ihrer VPC verwenden oder Ihr privates Netzwerk mit Ihrer VPC verbinden. Dies erreichen Sie mithilfe eines Amazon Virtual Private Network (VPN) oder mit AWS Direct Connect. Informationen zu Amazon VPN finden Sie unter [VPN-Verbindungen](#) im Benutzerhandbuch für Amazon Virtual Private Cloud. Weitere Informationen dazu AWS Direct Connect finden Sie unter [Verbindung erstellen](#) im AWS Direct Connect Benutzerhandbuch.

Sie können einen VPC-Schnittstellen-Endpunkt erstellen, um mithilfe der AWS Konsole oder der Befehle AWS Command Line Interface (AWS CLI) eine Verbindung zu Amazon EMR herzustellen. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#).

Wenn Sie nach dem Erstellen eines Schnittstellen-VPC-Endpunkts private DNS-Host-Namen für den Endpunkt aktivieren, wird der Standardendpunkt von Amazon EMR in den VPC-Endpunkt aufgelöst. Der Service-Standardname für den Endpunkt für Amazon EMR hat das folgende Format.

```
elasticmapreduce.Region.amazonaws.com
```

Wenn Sie keine privaten DNS-Hostnamen aktiviert haben, stellt Amazon VPC einen DNS-Endpunktnamen bereit, den Sie im folgenden Format verwenden können.

```
VPC_Endpoint_ID.elasticmapreduce.Region.vpce.amazonaws.com
```

Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon-VPC-Benutzerhandbuch.

Amazon EMR unterstützt Aufrufe aller [API-Aktionen](#) innerhalb Ihrer VPC.

Sie können VPC-Endpunktrichtlinien an einen VPC-Endpunkt anfügen, um den Zugriff für IAM-Prinzipale zu steuern. Sie können einem VPC-Endpunkt auch Sicherheitsgruppen zuordnen, um den eingehenden und ausgehenden Zugriff basierend auf Ursprung und Ziel des Netzwerkdatenverkehrs zu steuern, z. B. mit einem IP-Adressbereich. Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Services mit VPC-Endpunkten](#).

Eine VPC-Endpunktrichtlinie für Amazon EMR erstellen

Sie können eine Richtlinie für Amazon-VPC-Endpunkte für Amazon EMR erstellen, in der Sie Folgendes angeben:

- Prinzipal, der Aktionen ausführen/nicht ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Example — VPC-Endpunktrichtlinie, um jeglichen Zugriff von einem bestimmten AWS Konto aus zu verweigern

Die folgende VPC-Endpunktrichtlinie verweigert dem AWS Konto **123456789012** jeglichen Zugriff auf Ressourcen, die den Endpunkt verwenden.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Example – VPC-Endpunktrichtlinie zum Gewähren des VPC-Zugriffs auf einen angegebenen IAM-Prinzipal (Benutzer)

Die folgende VPC-Endpunktrichtlinie ermöglicht vollen Zugriff nur für das AWS Konto **123456789012** eines Benutzers **lijuan**. Allen anderen IAM-Prinzipalen wird der Zugriff über den Endpunkt verweigert.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/lijuan"
        ]
      }
    }
  ]
}
```

Example – VPC-Endpunktrichtlinie zum Erlauben von EMR-Leseoperationen

Die folgende VPC-Endpunktrichtlinie erlaubt nur AWS Konten **123456789012**, die angegebenen Amazon EMR-Aktionen durchzuführen.

Die angegebenen Aktionen stellen das Äquivalent von schreibgeschütztem Zugriff für Amazon EMR dar. Alle anderen Aktionen in der VPC werden dem angegebenen Konto verweigert. Allen anderen Konten wird der Zugriff verweigert. Eine Liste der Aktionen in Amazon EMR finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EMR](#).

```
{
  "Statement": [
    {
      "Action": [
        "elasticmapreduce:DescribeSecurityConfiguration",
        "elasticmapreduce:GetBlockPublicAccessConfiguration",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:ViewEventsFromAllClustersInConsole",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:ListInstanceFleets",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:DescribeStep",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:ListSecurityConfigurations",
        "elasticmapreduce:DescribeEditor",
        "elasticmapreduce:ListClusters",
        "elasticmapreduce:ListEditors"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Example – VPC-Endpunktrichtlinie, die den Zugriff auf einen angegebenen Cluster verweigert

Die folgende VPC-Endpunktrichtlinie ermöglicht vollen Zugriff für alle Konten und Principals, verweigert dem AWS Konto **123456789012** jedoch jeglichen Zugriff auf Aktionen, die auf dem Amazon EMR-Cluster mit Cluster-ID ausgeführt werden. **j-A1B2CD34EF5G** Andere Amazon-EMR-Aktionen, die keine Berechtigungen auf Ressourcenebene für Cluster unterstützen, sind weiterhin zulässig. Eine Liste der Amazon-EMR-Aktionen und die entsprechenden Ressourcentypen finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon EMR](#).

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "arn:aws:elasticmapreduce:us-west-2:123456789012:cluster/j-  
A1B2CD34EF5G",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```


Amazon EMR-Cluster verwalten

Nachdem Sie Ihren Cluster gestartet haben, können Sie eine Verbindung zu ihm herstellen und ihn verwalten. Amazon EMR bietet eine Sammlung von Tools, mit denen Sie dies tun können. In diesem Abschnitt finden Sie Anleitungen, wie Sie eine Verbindung zu Ihrem Cluster herstellen, ihm Aufgaben zuweisen, die Arbeit während der Ausführung überwachen und Probleme beheben können.

Themen

- [Eine Verbindung zu einem Amazon-EMR-Cluster herstellen](#)
- [Arbeit an einen Amazon EMR-Cluster einreichen](#)
- [Einen Amazon EMR-Cluster bei der Ausführung seiner Arbeit anzeigen und überwachen](#)
- [Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen](#)
- [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#)
- [Klonen Sie einen Amazon EMR-Cluster mithilfe der Konsole](#)
- [Automatisieren Sie wiederkehrende Amazon EMR-Cluster mit AWS Data Pipeline](#)

Eine Verbindung zu einem Amazon-EMR-Cluster herstellen

Wenn Sie einen Amazon-EMR-Cluster ausführen, müssen Sie häufig nur eine Anwendung ausführen, um die Daten zu analysieren, und dann die Ausgabe aus dem Amazon-S3-Bucket erfassen. In anderen Fällen möchten Sie vielleicht mit dem Primärknoten interagieren, während der Cluster ausgeführt wird. Sie möchten z. B. eine Verbindung mit dem Primärknoten herstellen, um interaktive Abfragen auszuführen, Protokolldateien zu prüfen, ein Problem mit dem Cluster zu debuggen, Leistung mithilfe einer Anwendung wie Ganglia zu überwachen, die im Primärknoten ausgeführt wird, und so weiter. In den folgenden Abschnitten werden Techniken beschrieben, mit denen Sie eine Verbindung mit dem Primärknoten herstellen können.

In einem EMR-Cluster ist der primäre Knoten eine EC2 Amazon-Instance, die die EC2 Instances koordiniert, die als Task- und Core-Knoten ausgeführt werden. Der Primärknoten stellt einen öffentlichen DNS-Namen bereit, den Sie verwenden können, um eine Verbindung mit der Instance herzustellen. Amazon EMR erstellt standardmäßig die Sicherheitsgruppenregeln für Primär-, Core- und Slave-Knoten, die bestimmen, wie Sie auf die Knoten zugreifen.

 Note

Eine Verbindung mit dem Primärknoten ist nur möglich, während der Cluster ausgeführt wird. Wenn der Cluster beendet wird, wird die EC2 Instance, die als primärer Knoten fungiert, beendet und ist nicht mehr verfügbar. Um eine Verbindung mit dem Primärknoten einzurichten, müssen Sie sich auch beim Cluster authentifizieren. Sie können entweder Kerberos für die Authentifizierung verwenden oder beim Starten des Clusters einen privaten EC2 Schlüssel für das Amazon-Schlüsselpaar angeben. Weitere Informationen zur Konfiguration von Kerberos und die Einrichtung einer Verbindung finden Sie unter [Verwenden Sie Kerberos für die Authentifizierung mit Amazon EMR](#). Wenn Sie einen Cluster von der Konsole aus starten, wird der private EC2 Schlüssel des Amazon-Schlüsselpaars im Abschnitt Sicherheit und Zugriff auf der Seite Cluster erstellen angegeben.

Standardmäßig erlaubt die Sicherheitsgruppe ElasticMapReduce -master keinen eingehenden SSH-Zugriff. Möglicherweise müssen Sie eine Regel hinzufügen, die eingehenden SSH-Zugriff (TCP-Port 22) von bestimmten Quellen erlaubt. Weitere Informationen zum Ändern von Sicherheitsgruppenregeln finden Sie unter [Regeln zu einer Sicherheitsgruppe hinzufügen](#) im EC2 Amazon-Benutzerhandbuch.

 Important

Ändern Sie die verbleibenden Regeln in der Sicherheitsgruppe ElasticMapReduce -master nicht. Das Ändern dieser Regeln kann die Ausführung des Clusters beeinträchtigen.

Themen

- [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#)
- [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#)

Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr

Bevor Sie eine Verbindung zu einem Amazon-EMR-Cluster herstellen, müssen Sie eingehenden SSH-Verkehr (Port 22) von vertrauenswürdigen Clients autorisieren, z. B. die IP-Adresse Ihres

Computers. Bearbeiten Sie dazu die verwalteten Sicherheitsgruppenregeln für die Knoten, zu denen Sie eine Verbindung herstellen möchten. Die folgenden Anweisungen zeigen Ihnen beispielsweise, wie Sie der Standard-Sicherheitsgruppe -Master eine Regel für eingehenden Datenverkehr für den SSH-Zugriff hinzufügen. ElasticMapReduce

Weitere Informationen zur Verwendung von Sicherheitsgruppen mit Amazon EMR finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Console

Um vertrauenswürdigen Quellen SSH-Zugriff auf die primäre Sicherheitsgruppe mit der Konsole zu gewähren

Um Ihre Sicherheitsgruppen zu bearbeiten, benötigen Sie die Berechtigung, Sicherheitsgruppen für die VPC zu verwalten, in der sich der Cluster befindet. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Ändern von Benutzerberechtigungen](#) und unter [Beispielrichtlinie](#), die die Verwaltung von EC2 Sicherheitsgruppen ermöglicht.

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters aus, und wählen Sie dann den Cluster aus, den Sie aktualisieren möchten. Dadurch wird die Cluster-Detailseite geöffnet. Die Registerkarte Eigenschaften auf dieser Seite sollte vorausgewählt sein.
3. Wählen Sie auf der Registerkarte Eigenschaften unter Netzwerk den Pfeil neben EC2 Sicherheitsgruppen (Firewall) aus, um diesen Abschnitt zu erweitern. Wählen Sie unter Primärknoten den Link zur Sicherheitsgruppe aus. Dadurch wird die EC2 Konsole geöffnet.
4. Wählen Sie die Registerkarte Eingehende Regeln und anschließend Eingehende Regeln bearbeiten aus.
5. Suchen Sie mit den folgenden Einstellungen nach einer Regel für eingehenden Datenverkehr, die öffentlichen Zugriff ermöglicht. Falls sie existiert, wählen Sie Löschen, um sie zu entfernen.

- Typ

SSH

- Port

22

- Quelle

Benutzerdefiniert 0.0.0.0/0

 Warning

Vor Dezember 2020 verfügte die Sicherheitsgruppe ElasticMapReduce -master über eine vorkonfigurierte Regel, die eingehenden Datenverkehr auf Port 22 aus allen Quellen zuließ. Diese Regel wurde erstellt, um die ersten SSH-Verbindungen zum Primärknoten zu vereinfachen. Wir empfehlen Ihnen dringend, diese Eingangsregel zu entfernen und den Datenverkehr auf vertrauenswürdige Quellen zu beschränken.

6. Scrollen Sie zum Ende der Regelliste und wählen Sie Regel hinzufügen.
7. Wählen Sie für Type (Typ) SSH aus. Bei dieser Auswahl werden automatisch TCP als Protokoll und 22 als Portbereich eingetragen.
8. Wählen Sie als Quelle Meine IP aus, um Ihre IP-Adresse automatisch als Quelladresse hinzuzufügen. Sie können auch einen Bereich benutzerdefinierter vertrauenswürdiger Client-IP-Adressen hinzufügen oder zusätzliche Regeln für andere Clients erstellen. In vielen Netzwerkumgebungen werden IP-Adressen dynamisch zugewiesen, sodass Sie in Zukunft möglicherweise Ihre IP-Adressen für vertrauenswürdige Clients aktualisieren müssen.
9. Wählen Sie Speichern.
10. Kehren Sie optional zu Schritt 3 zurück, wählen Sie Core- und Aufgabenknoten aus, und wiederholen Sie die Schritte 4 bis 8. Dadurch wird den Core- und Aufgabenknoten SSH-Clientzugriff gewährt.

Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her

Secure Shell (SSH) ist ein Netzwerkprotokoll, mit dem Sie eine sichere Verbindung mit einem Remote-Computer erstellen können. Nach dem Verbinden verhält sich das Terminal auf Ihrem lokalen Computer so, als würde es auf dem Remote-Computer ausgeführt. Lokal erstellte Befehle werden auf dem Remote-Computer ausgeführt und die Befehlsausgabe vom Remote-Computer wird im Terminal-Fenster angezeigt.

Wenn Sie SSH mit verwenden AWS, stellen Sie eine Verbindung zu einer EC2 Instance her, bei der es sich um einen virtuellen Server handelt, der in der Cloud ausgeführt wird. Bei der Arbeit mit Amazon EMR wird SSH am häufigsten verwendet, um eine Verbindung zu der EC2 Instance herzustellen, die als primärer Knoten des Clusters fungiert.

Wenn Sie SSH zum Herstellen einer Verbindung mit dem Primärknoten verwenden, können Sie den Cluster überwachen und mit ihm interagieren. Sie können Linux-Befehle auf dem Primärknoten absetzen, Anwendungen wie Hive und Pig interaktiv ausführen, Verzeichnisse durchsuchen, Protokolldateien lesen usw. Darüber hinaus können Sie einen Tunnel in Ihrer SSH-Verbindung erstellen, um die auf dem Primärknoten gehosteten Webschnittstellen anzeigen zu lassen.

Weitere Informationen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Um eine Verbindung mit dem Primärknoten unter Verwendung von SSH herzustellen, müssen Sie den öffentlichen DNS-Namen des Primärknotens verwenden. Darüber hinaus muss die dem Primärknoten zugeordnete Sicherheitsgruppe über eine Regel für eingehenden Datenverkehr verfügen, die SSH (TCP-Port 22)-Datenverkehr von einer Quelle zulässt, an der die SSH-Verbindung ihren Ursprung hat. Möglicherweise müssen Sie eine Regel hinzufügen, um eine SSH-Verbindung von Ihrem Client zuzulassen. Weitere Informationen zum Ändern von Sicherheitsgruppenregeln finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster Regeln zu einer Sicherheitsgruppe hinzufügen](#) im EC2 Amazon-Benutzerhandbuch.

Abrufen des öffentlichen DNS-Namens für den Primärknoten

Sie können den primären öffentlichen DNS-Namen über die Amazon-EMR-Konsole und die AWS CLI abrufen.

Console

So rufen Sie den öffentlichen DNS-Namen für den Primärknoten mit der neuen Konsole ab

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters aus, und wählen Sie dann den Cluster aus, von dem Sie den öffentlichen DNS-Namen abrufen möchten.
3. Beachten Sie den öffentlichen DNS-Wert des Primärknotens im Abschnitt Zusammenfassung der Cluster-Detailseite.

CLI

Um den öffentlichen DNS-Namen des primären Knotens abzurufen, verwenden Sie AWS CLI

1. Geben Sie den folgenden Befehl ein, um die Cluster-Kennung abzurufen:

```
aws emr list-clusters
```

In der Ausgabe werden Ihre Cluster einschließlich des Clusters aufgeführt IDs. Notieren Sie die Cluster-ID für den Cluster, mit dem Sie eine Verbindung herstellen.

```
"Status": {
  "Timeline": {
    "ReadyDateTime": 1408040782.374,
    "CreationDateTime": 1408040501.213
  },
  "State": "WAITING",
  "StateChangeReason": {
    "Message": "Waiting after step completed"
  }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "My cluster"
```

2. Geben Sie einen der folgenden Befehle ein, um die Cluster-Instances mitsamt dem öffentlichen DNS-Namen für den Cluster aufzulisten. `j-2AL4XXXXXX5T9` Ersetzen Sie durch die Cluster-ID, die vom vorherigen Befehl zurückgegeben wurde.

```
aws emr list-instances --cluster-id j-2AL4XXXXXX5T9
```

Oder:

```
aws emr describe-cluster --cluster-id j-2AL4XXXXXX5T9
```

Die Ausgabe enthält die Cluster-Instances einschließlich DNS-Namen und IP-Adressen. Notieren Sie den Wert für `PublicDnsName`.

```
"Status": {
  "Timeline": {
```

```

        "ReadyDateTime": 1408040779.263,
        "CreationDateTime": 1408040515.535
    },
    "State": "RUNNING",
    "StateChangeReason": {}
},
"Ec2InstanceId": "i-e89b45e7",
"PublicDnsName": "ec2-###-##-##-###.us-west-2.compute.amazonaws.com"

"PrivateDnsName": "ip-###-##-##-###.us-west-2.compute.internal",
"PublicIpAddress": "##.###.###.##",
"Id": "ci-12XXXXXXXXXXFMH",
"PrivateIpAddress": "###.##.#.###"

```

Weitere Informationen finden Sie unter [Amazon-EMR-Befehle in der AWS CLI](#).

Stellen Sie mithilfe von SSH und einem EC2 privaten Amazon-Schlüssel unter Linux, Unix und Mac OS X eine Connect zum Primärknoten her

Um eine SSH-Verbindung herzustellen, die mit einer privaten Schlüsseldatei authentifiziert wird, müssen Sie den privaten Schlüssel des EC2 Amazon-Schlüsselpaars angeben, wenn Sie einen Cluster starten. Weitere Informationen zum Zugriff auf Ihr key pair finden Sie unter [EC2 Amazon-Schlüsselpaare](#) im EC2 Amazon-Benutzerhandbuch.

Ihr Linux-Computer verfügt höchstwahrscheinlich standardmäßig über einen SSH-Client. OpenSSH wird beispielsweise bei den meisten Linux-, Unix- und MacOS-Betriebssystemen installiert. Sie können nach einem SSH-Client suchen, indem Sie in der Befehlszeile ssh eingeben. Wenn Ihr Computer den Befehl nicht erkennt, installieren Sie einen SSH-Client, um eine Verbindung mit dem Primärknoten herzustellen. Das OpenSSH-Projekt bietet eine kostenlose Implementierung der umfassenden Palette von SSH-Tools. Weitere Informationen finden Sie auf der [OpenSSH](#)-Website.

In den folgenden Anleitungen wird gezeigt, wie Sie eine SSH-Verbindung mit dem Amazon-EMR-Primärknoten unter Linux, Unix und Mac OS X öffnen.

So konfigurieren Sie Berechtigungen für die Datei mit dem privaten Schlüssel Ihres Schlüsselpaars

Bevor Sie Ihren privaten Schlüssel für das EC2 Amazon-Schlüsselpaar verwenden können, um eine SSH-Verbindung herzustellen, müssen Sie die Berechtigungen für die .pem Datei so einrichten, dass

nur der Schlüsselinhaber die Berechtigung hat, auf die Datei zuzugreifen. Dies ist erforderlich, um eine SSH-Verbindung mit dem Terminal oder dem herzustellen. AWS CLI

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).
2. Suchen Sie Ihre `.pem`-Datei. In dieser Anleitung wird davon ausgegangen, dass die Datei `mykeypair.pem` heißt und im Stammverzeichnis des aktuellen Benutzers gespeichert ist.
3. Geben Sie den folgenden Befehl ein, um die Berechtigungen festzulegen. `~/mykeypair.pem` Ersetzen Sie es durch den vollständigen Pfad und Dateinamen der privaten Schlüsseldatei Ihres Schlüsselpaars. Zum Beispiel `C:/Users/<username>/.ssh/mykeypair.pem`.

```
chmod 400 ~/mykeypair.pem
```

Wenn Sie keine Berechtigungen für die `.pem`-Datei festlegen, erhalten Sie die Fehlermeldung, dass Ihre Schlüsseldatei nicht geschützt ist und der Schlüssel abgelehnt wird. Zum Verbinden müssen Sie die Berechtigungen für die Datei mit dem privaten Schlüssel Ihres Schlüsselpaares nur bei der ersten Verwendung festlegen.

So stellen Sie eine Verbindung mit dem Primärknoten mithilfe des Terminals her

1. Öffnen Sie ein Terminal-Fenster. Wählen Sie unter Mac OS X Applications > Utilities > Terminal (Anwendungen > Dienstprogramme > Terminal) aus. In anderen Linux-Distributionen befindet sich „Terminal“ in der Regel unter Applications > Accessories > Terminal (Anwendungen > Zubehör > Terminal).
2. Geben Sie den folgenden Befehl ein, um eine Verbindung mit dem Primärknoten herzustellen. `ec2-###-##-##-###.compute-1.amazonaws.com` Ersetzen Sie es durch den primären öffentlichen DNS-Namen Ihres Clusters und `~/mykeypair.pem` ersetzen Sie es durch den vollständigen Pfad und Dateinamen Ihrer `.pem` Datei. Zum Beispiel `C:/Users/<username>/.ssh/mykeypair.pem`.

```
ssh hadoop@ec2-###-##-##-###.compute-1.amazonaws.com -i ~/mykeypair.pem
```


 Important

Sie müssen den Anmeldenamen `hadoop` verwenden, wenn Sie eine Verbindung mit dem Amazon-EMR-Primärknoten herstellen. Andernfalls wird eine Fehlermeldung ähnlich wie `Server refused our key` angezeigt.

3. Es wird die Warnung angezeigt, dass die Authentizität des Hosts, mit dem Sie eine Verbindung herstellen, nicht überprüft werden konnte. Geben Sie `yes` ein, um fortzufahren.
4. Wenn Sie Ihre Arbeit am Primärknoten abgeschlossen haben, geben Sie den folgenden Befehl ein, um die SSH-Verbindung zu schließen.

```
exit
```

Wenn Sie Probleme bei der Verwendung von SSH haben, um eine Verbindung zu Ihrem Primärknoten herzustellen, finden Sie weitere Informationen unter [Problembehandlung bei der Verbindung mit Ihrer Instance](#).

Verbinden mit dem Primärknoten mithilfe von SSH unter Windows

Windows-Benutzer können eine Verbindung mit dem Primärknoten mithilfe eines SSH-Clients wie z. B. PuTTY herstellen. Bevor Sie eine Verbindung zum Amazon EMR-Primärknoten herstellen, sollten Sie PuTTY und PuTTYgen herunterladen und installieren. TTYgen Sie können diese Tools auf der [PuTTY-Download-Seite](#) herunterladen.

PuTTY unterstützt das von Amazon generierte Schlüsselpaar-Dateiformat für private Schlüssel (`.pem`) nicht nativ. EC2 Sie verwenden PuTTYgen, um Ihre Schlüsseldatei in das erforderliche PuTTY-Format (`.ppk`) zu konvertieren. Sie müssen Ihren Schlüssel in dieses Format (`.ppk`) konvertieren, bevor Sie mithilfe von PuTTY eine Verbindung mit dem Primärknoten herstellen können.

Weitere Informationen zur Konvertierung Ihres Schlüssels finden Sie unter [Konvertieren Ihres privaten Schlüssels mit PuTTYgen](#) im EC2 Amazon-Benutzerhandbuch.

So stellen Sie eine Verbindung mit dem Primärknoten mithilfe von PuTTY her

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).
2. Öffnen Sie `putty.exe`. Sie können PuTTY auch über die Windows-Programmliste starten.
3. Falls erforderlich, wählen Sie in der Category (Kategorie)-Liste Session (Sitzung) aus.
4. Geben Sie als Hostname (oder IP-Adresse) ein `hadoop@MasterPublicDNS`. Beispiel: `hadoop@ec2-###-##-##-###.compute-1.amazonaws.com`.
5. Wählen Sie in der Category (Kategorie)-Liste Connection > SSH (Verbindung > SSH), Auth aus.
6. Klicken Sie bei Private key file for authentication (Private Schlüsseldatei für Authentifizierung) auf Browse (Durchsuchen), und wählen Sie die `.ppk`-Datei aus, die Sie generiert haben.
7. Wählen Sie Öffnen und dann Ja aus, um die PuTTY-Sicherheitswarnung zu schließen.

 Important

Wenn Sie sich beim Primärknoten anmelden und zur Angabe eines Benutzernamens aufgefordert werden, geben Sie `hadoop` ein.

8. Wenn Sie Ihre Arbeit am Primärknoten beendet haben, können Sie durch Schließen von PuTTY die SSH-Verbindung schließen.

 Note

Um zu verhindern, dass die SSH-Verbindung das Zeitlimit überschreitet, können Sie Connection (Verbindung) in der Category (Kategorie)-Liste und anschließend die Option Enable TCP_keepalives (TCP_keepalives aktivieren) auswählen. Wenn eine SSH-Sitzung in PuTTY aktiv ist, können Sie die Einstellungen ändern, indem Sie das Kontextmenü (rechte Maustaste) für die PuTTY-Titelleiste öffnen und dann Einstellungen ändern auswählen.

Wenn Sie Probleme bei der Verwendung von SSH haben, um eine Verbindung zu Ihrem Primärknoten herzustellen, finden Sie weitere Informationen unter [Problembehandlung bei der Verbindung mit Ihrer Instance](#).

Mit dem Primärknoten über die AWS CLI verbinden

Sie können unter Windows sowie unter Linux, Unix und Mac OS X eine SSH-Verbindung mit dem primären Knoten herstellen. Unabhängig von der Plattform benötigen Sie den öffentlichen DNS-Namen des primären Knotens und Ihren privaten Schlüssel für das EC2 Amazon-Schlüsselpaar. AWS CLI Wenn Sie den AWS CLI unter Linux, Unix oder Mac OS X verwenden, müssen Sie auch die Berechtigungen für die Datei mit dem privaten Schlüssel (.pem oder der Datei .ppk) festlegen, wie unter beschrieben. [So konfigurieren Sie Berechtigungen für die Datei mit dem privaten Schlüssel Ihres Schlüsselpaares](#)

Um eine Verbindung zum Primärknoten herzustellen, verwenden Sie AWS CLI

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).
2. Geben Sie Folgendes ein, um die Cluster-Kennung abzurufen:

```
aws emr list-clusters
```

In der Ausgabe werden Ihre Cluster einschließlich des Clusters aufgeführt IDs. Notieren Sie die Cluster-ID für den Cluster, mit dem Sie eine Verbindung herstellen.

```
"Status": {
  "Timeline": {
    "ReadyDateTime": 1408040782.374,
    "CreationDateTime": 1408040501.213
  },
  "State": "WAITING",
  "StateChangeReason": {
    "Message": "Waiting after step completed"
  }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "AWS CLI cluster"
```

3. Geben Sie den folgenden Befehl ein, um eine SSH-Verbindung mit dem Primärknoten zu öffnen. Ersetzen Sie im folgenden Beispiel durch **j-2AL4XXXXXX5T9** die Cluster-ID und **~/mykeypair.key** ersetzen Sie sie durch den vollständigen Pfad und Dateinamen Ihrer .pem

Datei (für Linux, Unix und Mac OS X) oder .ppk Datei (für Windows). Zum Beispiel C:\Users \<username>\.ssh\mykeypair.pem.

```
aws emr ssh --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

4. Wenn Sie mit der Arbeit am Primärknoten fertig sind, schließen Sie das AWS CLI Fenster.

Weitere Informationen finden Sie unter [Amazon-EMR-Befehle in der AWS CLI](#). Wenn Sie Probleme bei der Verwendung von SSH haben, um eine Verbindung zu Ihrem Primärknoten herzustellen, finden Sie weitere Informationen unter [Problembehandlung bei der Verbindung mit Ihrer Instance](#).

Amazon-EMR-Serviceports

 **Note**

Im Folgenden finden Sie Schnittstellen und Serviceports für Komponenten auf Amazon EMR. Dies ist keine vollständige Liste der Serviceports. Nicht standardmäßige Services, wie SSL-Ports und verschiedene Arten von Protokollen, sind nicht aufgeführt.

 **Important**

Seien Sie vorsichtig, wenn Sie Regeln für offene Ports für Sicherheitsgruppen bearbeiten. Stellen Sie sicher, dass Sie Regeln hinzufügen, die nur Datenverkehr von vertrauenswürdigen und authentifizierten Clients für die Protokolle und Ports zulassen, die zum Ausführen Ihrer Workloads erforderlich sind.

Komponente	Service description (Service-Beschreibung)	Der Service wird standardmäßig ausgeführt	Port	Schlüssel zur Konfiguration
Hadoop	HTTP-KMS-REST-API	Ja	9600	hadoop.kms.http.port

Komponente	Service description (Service-Beschreibung)	Der Service wird standardmäßig ausgeführt	Port	Schlüssel zur Konfiguration
HDFS	Namenode-Web-Benutzeroberfläche	Ja	9870	dfs.namenode.http-address
	Nameknode-RPC	Ja	8020	dfs.namenode.rpc-address
	DataNode Web-Benutzeroberfläche	Ja	9864	dfs.datanode.http-address
	Datanode-HTTP für die Datenübertragung	Ja	9866	dfs.datanode.address
	Datanode-RPC für die Datenübertragung	Ja	9867	dfs.datanode.ipc-address
Hive	HiveServer2 Sparsamkeit	Ja	10000	hive.server2.thrift.port
	HiveServer2 HTTP	Nein	10001	hive.server2.thrift.http.port
	HiveServer2 Weboberfläche	Ja	10002	hive.server2.webui.port
	Hive Metastore	Ja	9083	hive.metastore.port / metastore.thrift.port
	Netz HCat	Nein	50111	templeton.port

Komponente	Service description (Service-Beschreibung)	Der Service wird standardmäßig ausgeführt	Port	Schlüssel zur Konfiguration
	LLAP-Daemon-Verwaltungsservice (RPC)	Nein	15004	hive.llap.management.rpc.port
	YARN-Shuffle-Port für Shuffle LLAP-daemon-hosted	Nein	15551	hive.llap.daemon.yarn.shuffle.port
	Der LLAP-Daemon (RPC)	Nein	Dynamisch	hive.llap.daemon.rpc.port
	Webbenutzeroberfläche des LLAP-Daemons	Nein	15002	hive.llap.daemon.web.port
	LLAP-Daemon-Ausgabeservice	Nein	15003	hive.llap.daemon.output.service.port
Oozie		Ja	11000	
Tez	Tez UI	Ja	8080	
YARN	Shuffle	Ja	13562	mapreduce.shuffle.port
	Lokalisierer-RPC	Ja	8040	yarn.node.manager.localizer.address
		Ja	8041	

Komponente	Service description (Service-Beschreibung)	Der Service wird standardmäßig ausgeführt	Port	Schlüssel zur Konfiguration
	NM-Webapp-Adresse	Ja	8042	yarn.node.manager.webapp.address
	RM-Webanwendung	Ja	8088	yarn.resourcemanager.webapp.address
		Ja	8025	
	Scheduler	Ja	8030	yarn.resourcemanager.scheduler.address
	Schnittstelle für den Anwendungsmanager	Ja	8032	yarn.resourcemanager.address
	RM-Admin-Oberfläche	Ja	8033	yarn.resourcemanager.admin.address
	JobHistory Web-Benutzeroberfläche des Servers	Ja	1988	mapreduce.jobhistory.webapp.address
	JobHistory Webbenutzeroberfläche für Serveradministratoren	Ja	10033	mapreduce.jobhistory.admin.address

Komponente	Service description (Service-Beschreibung)	Der Service wird standardmäßig ausgeführt	Port	Schlüssel zur Konfiguration
	JobHistory Server (RPC)	Ja	10020	mapreduce.jobhistory.address
	Anwendungs-Timeline-Server (RPC)	Ja	10200	garn.timeline-service.adresse
	HTTP-Webbenutzeroberfläche für Anwendungsszeitserver	Ja	8188	garn.timeline-service.webapp.adresse
	HTTPS-Webbenutzeroberfläche für Anwendungsszeitserver	Nein	8190	yarn.timeline-service.webapp.https.address
		Ja	20888	
Zookeeper	Client-Port	Ja	2181	
		Ja	37301	
		Ja	8341	

Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen

Important

Sie können eine benutzerdefinierte Sicherheitsgruppe konfigurieren, um den eingehenden Zugriff auf diese Webschnittstellen zu ermöglichen. Beachten Sie, dass jeder Port, an dem Sie eingehenden Datenverkehr zulassen, eine potenzielle Sicherheitslücke darstellt.

Überprüfen Sie sorgfältig die benutzerdefinierten Sicherheitsgruppen, um Schwachstellen zu minimieren. Weitere Informationen finden Sie unter [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Hadoop und andere Anwendungen, die Sie auf Ihrem -Cluster installieren, veröffentlichen Benutzeroberflächen als auf dem Primärknoten gehostete Websites. Aus Sicherheitsgründen stehen diese Websites bei der Verwendung von Amazon EMR verwalteten Sicherheitsgruppen nur auf dem lokalen Webserver des Primärknotens zur Verfügung. Daher müssen Sie eine Verbindung mit dem Primärknoten herstellen, um sie anzuzeigen. Aus diesem Grund müssen Sie eine Verbindung zum Primärknoten herstellen, um die Weboberflächen anzeigen zu können. Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#). Hadoop veröffentlicht Benutzeroberflächen auch als Websites, die auf Core- und Aufgabenknoten gehostet werden. Diese Websites sind ebenfalls nur auf dem lokalen Webserver auf dem Knoten verfügbar.

Die folgende Tabelle enthält die Webschnittstellen, die Sie auf Cluster-Instances anzeigen lassen können: Diese Hadoop-Schnittstellen sind in allen Clustern verfügbar. Ersetzen Sie die Master-Instance-Schnittstellen *master-public-dns-name* durch das öffentliche Master-DNS, das auf der Registerkarte Cluster-Zusammenfassung in der Amazon EMR-Konsole aufgeführt ist. Bei Core- und Task-Instance-Schnittstellen *coretask-public-dns-name* ersetzen Sie diesen durch den öffentlichen DNS-Namen, der für die Instance aufgeführt ist. Um den Öffentlicher DNS-Name einer instance zu finden, wählen Sie in der Amazon-EMR-Konsole Ihren Cluster aus der Liste, wählen Sie die Registerkarte Hardware, wählen Sie die ID der Instance-Gruppe, die die Instance enthält, zu der Sie eine Verbindung herstellen möchten, und notieren Sie sich dann den für die Instance angegebenen Öffentlicher DNS-Name.

Name der Schnittstelle	URI
Flink History Server (EMR-Version 5.33 und höher)	http: //:8082/ <i>master-public-dns-name</i>
Ganglia	http:///ganglia/ <i>master-public-dns-name</i>
Hadoop HDFS NameNode (EMR-Version vor 6.x)	https: <i>master-public-dns-name</i> //50470/

Name der Schnittstelle	URI
Hadoop HDFS NameNode	http: //:50070/ <i>master-public-dns-name</i>
Hadoop HDFS DataNode	http: //:50075/ <i>coretask-public-dns-name</i>
Hadoop HDFS NameNode (EMR-Version 6.x)	https: <i>master-public-dns-name</i> //:9870/
Hadoop HDFS DataNode (EMR-Version vor 6.x)	https: <i>coretask-public-dns-name</i> //50475/
Hadoop HDFS DataNode (EMR-Version 6.x)	https: <i>coretask-public-dns-name</i> //9865/
HBase	http: //:16010/ <i>master-public-dns-name</i>
Hue	http: //:8888/ <i>master-public-dns-name</i>
JupyterHub	https: //:9443/ <i>master-public-dns-name</i>
Livy	http: //:8998/ <i>master-public-dns-name</i>
Funke HistoryServer	http: //:18080/ <i>master-public-dns-name</i>
Tez	http: //:8080/tez-ui <i>master-public-dns-name</i>
GARN NodeManager	http: //:8042/ <i>coretask-public-dns-name</i>
GARN ResourceManager	http: //:8088/ <i>master-public-dns-name</i>
Zeppelin	http: //:8890/ <i>master-public-dns-name</i>

Da mehrere anwendungsspezifische Schnittstellen für den Primärknoten, aber nicht für den Core- und Aufgabenknoten verfügbar sind, gelten die Anweisungen in diesem Dokument speziell für den Amazon-EMR-Primärknoten. Auf die Webschnittstellen im Core- und Aufgabenknoten kann auf die gleiche Weise zugegriffen werden wie auf die Webschnittstellen im Primärknoten.

Es gibt mehrere Möglichkeiten, auf die Webschnittstellen im Primärknoten zuzugreifen. Am einfachsten und schnellsten stellen Sie eine Verbindung mit dem Primärknoten mithilfe von SSH her und verwenden den Browser Lynx zum Betrachten der Websites in Ihrem SSH-Client. Lynx ist jedoch ein textbasierter Browser mit einer eingeschränkten Benutzeroberfläche, die keine Grafiken anzeigen kann. Das folgende Beispiel zeigt, wie die ResourceManager Hadoop-Schnittstelle mit Lynx geöffnet wird (Lynx URLs werden auch bereitgestellt, wenn Sie sich mit SSH am primären Knoten anmelden).

```
lynx http://ip-###-##-###.us-west-2.compute.internal:8088/
```

Es gibt zwei verbleibende Optionen für den Zugriff auf Webschnittstellen im Primärknoten, die vollständige Browserfunktionalität bieten. Wählen Sie eine der folgenden Optionen aus:

- Option 1 (empfohlen für technisch fortgeschrittene Benutzer): Stellen Sie eine Verbindung mit dem Primärknoten mithilfe eines SSH-Clients her, konfigurieren Sie SSH-Tunneling mit lokaler Port-Weiterleitung und verwenden Sie einen Internetbrowser, um auf dem Primärknoten gehostete Webschnittstellen zu öffnen. Durch diese Methode können Sie den Zugriff auf Webschnittstellen so konfigurieren, dass kein SOCKS-Proxy benötigt wird.
- Option 2 (für neue Benutzer empfohlen): Verwenden Sie einen SSH-Client, um eine Verbindung zum primären Knoten herzustellen, konfigurieren Sie SSH-Tunneling mit dynamischer Portweiterleitung und konfigurieren Sie Ihren Internetbrowser so, dass er ein Add-on wie FoxyProxy für Firefox oder SwitchyOmega Chrome verwendet, um Ihre SOCKS-Proxyeinstellungen zu verwalten. Mit dieser Methode können Sie automatisch URLs anhand von Textmustern filtern und die Proxyeinstellungen auf Domänen beschränken, die der Form des DNS-Namens des primären Knotens entsprechen. Weitere Informationen zur Konfiguration FoxyProxy für Firefox und Google Chrome finden Sie unter [Option 2, Teil 2: Proxyeinstellungen konfigurieren, um Websites anzuzeigen, die auf dem primären Knoten des Amazon EMR-Clusters gehostet werden](#).

Note

Wenn Sie den Port, auf dem eine Anwendung ausgeführt wird, über die Cluster-Konfiguration ändern, wird der Hyperlink zum Port in der Amazon-EMR-Konsole nicht aktualisiert. Das liegt daran, dass die Konsole nicht über die Funktionalität verfügt, die Konfiguration `server.port` zu lesen.

Mit Amazon EMR Version 5.25.0 oder höher können Sie über die Konsole auf die Benutzeroberfläche des Spark History Servers zugreifen, ohne einen Web-Proxy über eine SSH-Verbindung einzurichten.

Weitere Informationen finden Sie unter [Zugriff auf den persistenten Spark History Server mit nur einem Klick](#).

Themen

- [Option 1: Richten Sie mithilfe der lokalen Portweiterleitung einen SSH-Tunnel zum Amazon EMR-Primärknoten ein](#)
- [Option 2, Teil 1: Einen SSH-Tunnel zum Primärknoten mithilfe der dynamischen Port-Weiterleitung einrichten](#)
- [Option 2, Teil 2: Proxyeinstellungen konfigurieren, um Websites anzuzeigen, die auf dem primären Knoten des Amazon EMR-Clusters gehostet werden](#)

Option 1: Richten Sie mithilfe der lokalen Portweiterleitung einen SSH-Tunnel zum Amazon EMR-Primärknoten ein

Stellen Sie eine Verbindung mit dem lokalen Webserver im Primärknoten her, indem Sie einen SSH-Tunnel zwischen Ihrem Computer und dem Primärknoten erstellen. Dies wird auch als Port-Weiterleitung bezeichnet. Wenn Sie keinen SOCKS-Proxy verwenden möchten, können Sie einen SSH-Tunnel zum Primärknoten mithilfe der lokalen Port-Weiterleitung einrichten. Bei der lokalen Port-Weiterleitung geben Sie ungenutzte lokale Ports an, die zum Weiterleiten von Datenverkehr zu bestimmten Remote-Ports auf dem lokalen Webserver des Primärknotens verwendet werden.

Zum Einrichten eines SSH-Tunnels mithilfe der lokalen Port-Weiterleitung benötigen Sie den öffentlichen DNS-Namen des Primärknotens und die Datei mit dem privaten Schlüssel Ihres Schlüsselpaares. Weitere Informationen darüber, wie Sie den öffentlichen DNS-Namen für den Master abrufen, finden Sie unter [Abrufen des öffentlichen DNS-Namens für den Primärknoten](#). Weitere Informationen zum Zugriff auf Ihr key pair finden Sie unter [EC2 Amazon-Schlüsselpaare](#) im EC2 Amazon-Benutzerhandbuch. Weitere Informationen zu den Websites, die Sie sich auf dem Primärknoten ansehen können, finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Einen SSH-Tunnel zum Primärknoten mithilfe der lokalen Port-Weiterleitung unter Windows einrichten

So richten Sie einen SSH-Tunnel mithilfe der lokalen Port-Weiterleitung im Terminal ein

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).

2. Öffnen Sie ein Terminal-Fenster. Wählen Sie unter Mac OS X Applications > Utilities > Terminal (Anwendungen > Dienstprogramme > Terminal) aus. In anderen Linux-Distributionen befindet sich „Terminal“ in der Regel unter Applications > Accessories > Terminal (Anwendungen > Zubehör > Terminal).
3. Geben Sie den folgenden Befehl ein, um einen SSH-Tunnel auf Ihrem lokalen Computer zu öffnen. Dieser Beispielbefehl greift auf die ResourceManager Weboberfläche zu, indem er den Datenverkehr auf dem lokalen Port 8157 (einem zufällig ausgewählten ungenutzten lokalen Port) an den Port 8088 auf dem lokalen Webserver des Master-Knotens weiterleitet.

Ersetzen Sie den Befehl `~/mykeypair.pem` durch den Speicherort und den Dateinamen Ihrer .pem Datei und `ec2-###-##-##-###.compute-1.amazonaws.com` ersetzen Sie ihn durch den öffentlichen Master-DNS-Namen Ihres Clusters. Um auf eine andere Weboberfläche zuzugreifen, 8088 ersetzen Sie diese durch die entsprechende Portnummer. Ersetzen Sie beispielsweise 8088 durch 8890 für die Zeppelin-Schnittstelle.

```
ssh -i ~/mykeypair.pem -N -L 8157:ec2-###-##-##-###.compute-1.amazonaws.com:8088 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
```

-L bezeichnet die Verwendung der lokalen Port-Weiterleitung. Damit können Sie einen lokalen Port für die Weiterleitung von Datenverkehr zu einem bestimmten Remote-Port auf dem lokalen Webserver des Hauptknotens angeben.

Nachdem Sie diesen Befehl ausgeführt haben, bleibt das Terminal geöffnet und gibt keine Antwort zurück.

4. Um die ResourceManager Weboberfläche in Ihrem Browser zu öffnen, geben Sie `http://localhost:8157/` in die Adressleiste ein.
5. Wenn Sie die Arbeit mit den Webschnittstellen im Primärknoten beendet haben, schließen Sie die Terminal-Fenster.

Option 2, Teil 1: Einen SSH-Tunnel zum Primärknoten mithilfe der dynamischen Port-Weiterleitung einrichten

Stellen Sie eine Verbindung mit dem lokalen Webserver im Primärknoten her, indem Sie einen SSH-Tunnel zwischen Ihrem Computer und dem Primärknoten erstellen. Dies wird auch als Port-Weiterleitung bezeichnet. Wenn Sie Ihren SSH-Tunnel mithilfe der dynamischen Port-Weiterleitung erstellen, wird der gesamte, an einen bestimmten ungenutzten lokalen Port geleitete Datenverkehr

an den lokalen Web-Server des Primärknotens weitergeleitet. Dies erstellt einen SOCKS-Proxy. Anschließend können Sie Ihren Internetbrowser so konfigurieren, dass er ein Add-On verwendet FoxyProxy oder SwitchyOmega Ihre SOCKS-Proxyeinstellungen verwaltet.

Mithilfe eines Proxy-Management-Add-ons können Sie automatisch URLs anhand von Textmustern filtern und die Proxyeinstellungen auf Domänen beschränken, die der Form des öffentlichen DNS-Namens des primären Knotens entsprechen. Das Browser-Add-On aktiviert und deaktiviert den Proxy automatisch, wenn Sie zwischen den auf dem Primärknoten gehosteten Websites und solchen im Internet wechseln.

Bevor Sie beginnen, benötigen Sie den öffentlichen DNS-Namen des Primärknotens und die Datei mit dem privaten Schlüssel Ihres Schlüsselpaares. Informationen darüber, wie Sie den primären öffentlichen DNS-Namen finden, erhalten Sie unter [Abrufen des öffentlichen DNS-Namens für den Primärknoten](#). Weitere Informationen zum Zugriff auf Ihr key pair finden Sie unter [EC2 Amazon-Schlüsselpaare](#) im EC2 Amazon-Benutzerhandbuch. Weitere Informationen zu den Websites, die Sie sich auf dem Primärknoten ansehen können, finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Richten Sie mithilfe der dynamischen Portweiterleitung mit OpenSSH einen SSH-Tunnel zum Primärknoten ein

So richten Sie einen SSH-Tunnel mithilfe der dynamischen Portweiterleitung mit OpenSSH ein

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).
2. Öffnen Sie ein Terminal-Fenster. Wählen Sie unter Mac OS X Applications > Utilities > Terminal (Anwendungen > Dienstprogramme > Terminal) aus. In anderen Linux-Distributionen befindet sich „Terminal“ in der Regel unter Applications > Accessories > Terminal (Anwendungen > Zubehör > Terminal).
3. Geben Sie den folgenden Befehl ein, um einen SSH-Tunnel auf Ihrem lokalen Computer zu öffnen. `~/mykeypair.pem` Ersetzen Sie es durch den Speicherort und den Dateinamen Ihrer .pem Datei, `8157` ersetzen Sie es durch eine unbenutzte, lokale Portnummer und `ec2-###-##-###-###.compute-1.amazonaws.com` ersetzen Sie es durch den primären öffentlichen DNS-Namen Ihres Clusters.

```
ssh -i ~/mykeypair.pem -N -D 8157 hadoop@ec2-###-##-###-###.compute-1.amazonaws.com
```

Nachdem Sie diesen Befehl ausgeführt haben, bleibt das Terminal geöffnet und gibt keine Antwort zurück.

 Note

-D bezeichnet die Verwendung der dynamischen Port-Weiterleitung. Damit können Sie einen lokalen Port für die Weiterleitung von Datenverkehr zu allen Remote-Ports auf dem lokalen Webserver des Primärknotens angeben. Die dynamische Port-Weiterleitung erstellt einen lokalen SOCKS-Proxy, der den im Befehl angegebenen Port überwacht.

4. Wenn der Tunnel aktiv ist, konfigurieren Sie einen SOCKS-Proxy für Ihren Browser. Weitere Informationen finden Sie unter [Option 2, Teil 2: Proxyeinstellungen konfigurieren, um Websites anzuzeigen, die auf dem primären Knoten des Amazon EMR-Clusters gehostet werden](#).
5. Wenn Sie die Arbeit mit den Webschnittstellen im Primärknoten beendet haben, schließen Sie das Terminal-Fenster.

Richten Sie einen SSH-Tunnel mit dynamischer Portweiterleitung mit dem ein AWS CLI

Sie können unter Windows sowie unter Linux, Unix und Mac OS X eine SSH-Verbindung mit dem Primärknoten herstellen. Wenn Sie den AWS CLI unter Linux, Unix oder Mac OS X verwenden, müssen Sie die Berechtigungen für die `.pem` Datei festlegen, wie unter beschrieben. AWS CLI [So konfigurieren Sie Berechtigungen für die Datei mit dem privaten Schlüssel Ihres Schlüsselpaares](#) Wenn Sie das AWS CLI unter Windows verwenden, muss PuTTY in der Umgebungsvariablen `path` erscheinen. Andernfalls wird möglicherweise eine Fehlermeldung wie OpenSSH oder PuTTY nicht verfügbar angezeigt.

Um einen SSH-Tunnel mit dynamischer Portweiterleitung einzurichten, verwenden Sie AWS CLI

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).
2. Erstellen Sie eine SSH-Verbindung mit dem Primärknoten wie unter [Mit dem Primärknoten über die AWS CLI verbinden](#) gezeigt.
3. Geben Sie Folgendes ein, um die Cluster-Kennung abzurufen:

```
aws emr list-clusters
```

In der Ausgabe werden Ihre Cluster einschließlich des Clusters IDs aufgeführt. Notieren Sie die Cluster-ID für den Cluster, mit dem Sie eine Verbindung herstellen.

```
"Status": {
  "Timeline": {
    "ReadyDateTime": 1408040782.374,
    "CreationDateTime": 1408040501.213
  },
  "State": "WAITING",
  "StateChangeReason": {
    "Message": "Waiting after step completed"
  }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "AWS CLI cluster"
```

4. Geben Sie den folgenden Befehl ein, um einen SSH-Tunnel zum Primärknoten mithilfe der dynamischen Port-Weiterleitung zu öffnen. Ersetzen Sie im folgenden Beispiel durch **j-2AL4XXXXXX5T9** die Cluster-ID und **~/mykeypair.key** ersetzen Sie sie durch den Speicherort und den Dateinamen Ihrer .pem Datei (für Linux, Unix und Mac OS X) oder .ppk Datei (für Windows).

```
aws emr socks --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

Note

Mit dem Socks-Befehl wird die dynamische Port-Weiterleitung am lokalen Port 8157 automatisch konfiguriert. Derzeit kann diese Einstellung nicht geändert werden.

5. Wenn der Tunnel aktiv ist, konfigurieren Sie einen SOCKS-Proxy für Ihren Browser. Weitere Informationen finden Sie unter [Option 2, Teil 2: Proxyeinstellungen konfigurieren, um Websites anzuzeigen, die auf dem primären Knoten des Amazon EMR-Clusters gehostet werden](#).
6. Wenn Sie mit der Arbeit mit den Webschnittstellen auf dem primären Knoten fertig sind, schließen Sie das AWS CLI Fenster.

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Mit PuTTY einen SSH-Tunnel zum Primärknoten einrichten

Windows-Benutzer können einen SSH-Tunnel zum Primärknoten mithilfe eines SSH-Clients wie z. B. PuTTY erstellen. Bevor Sie eine Verbindung zum Amazon EMR-Primärknoten herstellen, sollten Sie PuTTY und Pu herunterladen und installieren. TTYgen Sie können diese Tools auf der [PuTTY-Download-Seite](#) herunterladen.

PuTTY unterstützt das von Amazon generierte Schlüsselpaar-Dateiformat für private Schlüssel (.pem) nicht nativ. EC2 Sie verwenden PuTTYgen, um Ihre Schlüsseldatei in das erforderliche PuTTY-Format (.ppk) zu konvertieren. Sie müssen Ihren Schlüssel in dieses Format (.ppk) konvertieren, bevor Sie mithilfe von PuTTY eine Verbindung mit dem Primärknoten herstellen können.

Weitere Informationen zur Konvertierung Ihres Schlüssels finden Sie unter [Konvertieren Ihres privaten Schlüssels mit PuTTYgen](#) im EC2 Amazon-Benutzerhandbuch.

So richten Sie einen SSH-Tunnel mit dynamischer Portweiterleitung mithilfe von PuTTY ein

1. Stellen Sie sicher, dass Sie eingehenden SSH-Verkehr zugelassen haben. Detaillierte Anweisungen finden Sie unter [Bevor Sie eine Verbindung zu Amazon EMR herstellen: Autorisieren Sie eingehenden Datenverkehr](#).
2. Doppelklicken Sie auf `putty.exe`, um PuTTY zu starten. Sie können PuTTY auch über die Windows-Programmliste starten.

Note

Wenn bereits eine aktive SSH-Sitzung mit dem Primärknoten vorhanden ist, können Sie einen Tunnel hinzufügen. Klicken Sie hierfür mit der rechten Maustaste auf die PuTTY-Titelleiste und wählen Sie `Einstellungen ändern` aus.

3. Falls erforderlich, wählen Sie in der Category (Kategorie)-Liste Session (Sitzung) aus.
4. Geben Sie in das Feld Hostname ein **hadoopemasterpublicdns**. Beispiel: **hadoopemasterpublicdns-###-##-##-###.compute-1.amazonaws.com**.
5. Erweitern Sie in der Liste Category (Kategorie) die Option Connection > SSH (Verbindung > SSH), und wählen Sie anschließend Auth aus.
6. Klicken Sie bei Private key file for authentication (Private Schlüsseldatei für Authentifizierung) auf Browse (Durchsuchen), und wählen Sie die .ppk-Datei aus, die Sie generiert haben.

 Note

PuTTY unterstützt das von Amazon generierte Schlüsselpaar-Dateiformat für private Schlüssel (.pem) nicht nativ. EC2 Sie verwenden PuTTYgen , um Ihre Schlüsseldatei in das erforderliche PuTTY-Format (.ppk) zu konvertieren. Sie müssen Ihren Schlüssel in dieses Format (.ppk) konvertieren, bevor Sie mithilfe von PuTTY eine Verbindung mit dem Primärknoten herstellen können.

7. Erweitern Sie in der Liste Category (Kategorie) die Option Connection > SSH (Verbindung (SSH)), und wählen Sie anschließend Tunnels (Tunnel) aus.
8. Geben Sie im Feld Quellport 8157 (einen nicht verwendeten lokalen Port) ein und wählen Sie dann Hinzufügen aus.
9. Lassen Sie das Feld Destination (Zieladresse) leer.
10. Wählen Sie die Optionen Dynamic (Dynamisch) und Auto.
11. Klicken Sie auf Open.
12. Wählen Sie Yes (Ja) aus, um die PuTTY-Sicherheitswarnung zu schließen.

 Important

Wenn Sie sich beim Primärknoten anmelden, geben Sie hadoop ein, wenn Sie zur Angabe eines Benutzernamens aufgefordert werden.

13. Wenn der Tunnel aktiv ist, konfigurieren Sie einen SOCKS-Proxy für Ihren Browser. Weitere Informationen finden Sie unter [Option 2, Teil 2: Proxyeinstellungen konfigurieren, um Websites anzuzeigen, die auf dem primären Knoten des Amazon EMR-Clusters gehostet werden](#).
14. Wenn Sie die Arbeit mit den Webschnittstellen im Primärknoten beendet haben, schließen Sie das PuTTY-Fenster.

Option 2, Teil 2: Proxyeinstellungen konfigurieren, um Websites anzuzeigen, die auf dem primären Knoten des Amazon EMR-Clusters gehostet werden

Wenn Sie einen SSH-Tunnel mit dynamischer Port-Weiterleitung verwenden, müssen Sie ein Add-On für die SOCKS-Proxy-Verwaltung einsetzen, um die Proxy-Einstellungen in Ihrem Browser zu steuern. Mithilfe eines SOCKS-Proxy-Management-Tools können Sie automatisch URLs anhand von Textmustern filtern und die Proxyeinstellungen auf Domänen beschränken, die der Form des

öffentlichen DNS-Namens des primären Knotens entsprechen. Das Browser-Add-On aktiviert und deaktiviert den Proxy automatisch, wenn Sie zwischen den auf dem Primärknoten gehosteten Websites und solchen im Internet wechseln. Um Ihre Proxyeinstellungen zu verwalten, konfigurieren Sie Ihren Browser so, dass er ein Add-on wie FoxyProxy oder verwendet SwitchyOmega.

Weitere Informationen zum Erstellen eines SSH-Tunnels finden Sie unter [Option 2, Teil 1: Einen SSH-Tunnel zum Primärknoten mithilfe der dynamischen Port-Weiterleitung einrichten](#). Weitere Informationen zu den verfügbaren Webschnittstellen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Geben Sie bei der Einrichtung Ihres Proxy-Add-ons die folgenden Einstellungen an:

- Verwenden Sie localhost als Hostadresse.
- Verwenden Sie dieselbe lokale Portnummer, die Sie für die Einrichtung des SSH-Tunnels mit dem Primärknoten ausgewählt haben. [Option 2, Teil 1: Einen SSH-Tunnel zum Primärknoten mithilfe der dynamischen Port-Weiterleitung einrichten](#) Zum Beispiel Port~~8~~**157**. Dieser Port muss zudem mit der Port-Nummer in PuTTY oder anderen Terminal-Emulatoren übereinstimmen, die Sie zum Verbinden nutzen.
- Geben Sie das SOCKS-v5-Protokoll an. Mit SOCKS v5 können Sie optional die Benutzerautorisierung einrichten.
- URL Patterns (URL-Muster)

Die folgenden URL-Muster sollten in die Zulassungsliste aufgenommen und mit einem Platzhaltermustertyp angegeben werden:

- Die Muster `*ec2*.compute*.amazonaws.com*` und `*10*.amazonaws.com*` stimmen mit den öffentlichen DNS-Namen von Clustern in US-Regionen überein.
- Die Muster `*ec2*.compute*` und `*10*.compute*` stimmen mit den öffentlichen DNS-Namen von Clustern in allen anderen Regionen überein.
- Eine 10. * Muster für den Zugriff auf die JobTracker Protokolldateien in Hadoop. Ändern Sie diesen Filter bei Konflikten mit Ihrem Netzwerkzugriffsplan.
- Die Muster `*.ec2.internal*` und `*.compute.internal*` müssen den privaten (internen) DNS-Namen der Cluster in der `us-east-1`-Region bzw. allen anderen Regionen entsprechen.

Beispiel: FoxyProxy Für Firefox konfigurieren

Das folgende Beispiel zeigt eine FoxyProxy Standardkonfiguration (Version 7.5.1) für Mozilla Firefox.

FoxyProxy stellt eine Reihe von Tools zur Proxyverwaltung bereit. Damit können Sie einen Proxy-Server verwenden, um Muster abzugleichen URLs, die den Domänen entsprechen, die von den EC2 Amazon-Instances in Ihrem Amazon EMR-Cluster verwendet werden.

Zur Installation und Konfiguration FoxyProxy mit Mozilla Firefox

1. Gehen Sie in Firefox zu <https://addons.mozilla.org/>, suchen Sie nach FoxyProxy Standard und folgen Sie den Anweisungen zum Hinzufügen FoxyProxy zu Firefox.
2. Erstellen Sie mit einem Texteditor eine JSON-Datei mit dem Namen `foxyproxy-settings.json` aus der folgenden Beispielkonfiguration.

```
{
  "k20d21508277536715": {
    "active": true,
    "address": "localhost",
    "port": 8157,
    "username": "",
    "password": "",
    "type": 3,
    "proxyDNS": true,
    "title": "emr-socks-proxy",
    "color": "#0055E5",
    "index": 9007199254740991,
    "whitePatterns": [
      {
        "title": "*ec2*.compute*.amazonaws.com*",
        "active": true,
        "pattern": "*ec2*.compute*.amazonaws.com*",
        "importedPattern": "*ec2*.compute*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*ec2*.compute*",
        "active": true,
        "pattern": "*ec2*.compute*",
        "importedPattern": "*ec2*.compute*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "10.*",
```

```

    "active": true,
    "pattern": "10.*",
    "importedPattern": "http://10.*",
    "type": 1,
    "protocols": 2
  },
  {
    "title": "*10*.amazonaws.com*",
    "active": true,
    "pattern": "*10*.amazonaws.com*",
    "importedPattern": "*10*.amazonaws.com*",
    "type": 1,
    "protocols": 1
  },
  {
    "title": "*10*.compute*",
    "active": true,
    "pattern": "*10*.compute*",
    "importedPattern": "*10*.compute*",
    "type": 1,
    "protocols": 1
  },
  {
    "title": "/*.compute.internal*",
    "active": true,
    "pattern": "/*.compute.internal*",
    "importedPattern": "/*.compute.internal*",
    "type": 1,
    "protocols": 1
  },
  {
    "title": "/*.ec2.internal* ",
    "active": true,
    "pattern": "/*.ec2.internal*",
    "importedPattern": "/*.ec2.internal*",
    "type": 1,
    "protocols": 1
  }
],
"blackPatterns": [],
"logging": {
  "size": 100,
  "active": false
}

```

```
},  
  "mode": "patterns",  
  "browserVersion": "68.12.0",  
  "foxyProxyVersion": "7.5.1",  
  "foxyProxyEdition": "standard"  
}
```

3. Öffnen Sie die Firefox-Seite Ihre Erweiterungen verwalten (gehen Sie zu `about:addons` und wählen Sie dann Erweiterungen).
4. Wählen Sie FoxyProxy Standard und dann die Schaltfläche für weitere Optionen (die Schaltfläche, die wie eine Ellipse aussieht).
5. Wählen Sie Optionen aus der Dropdown-Liste aus.
6. Wählen Sie im linken Menü die Option Einstellungen importieren.
7. Wählen Sie auf der Seite „Importeinstellungen“ unter „Importeinstellungen aus Version FoxyProxy 6.0+“ die Option „Importeinstellungen“, navigieren Sie zum Speicherort der von Ihnen erstellten **foxyproxy-settings.json** Datei, wählen Sie die Datei aus und wählen Sie „Öffnen“.
8. Wählen Sie OK, wenn Sie aufgefordert werden, die vorhandenen Einstellungen zu überschreiben und Ihre neue Konfiguration zu speichern.

Beispiel: Für Chrome konfigurieren SwitchyOmega

Das folgende Beispiel zeigt, wie Sie die SwitchyOmega Erweiterung für Google Chrome einrichten. SwitchyOmega ermöglicht es Ihnen, mehrere Proxys zu konfigurieren, zu verwalten und zwischen ihnen zu wechseln.

Zur Installation und Konfiguration SwitchyOmega mit Google Chrome

1. Gehen Sie zu <https://chrome.google.com/webstore/Kategorie/Erweiterungen>, suchen Sie nach Proxy SwitchyOmega und fügen Sie ihn zu Chrome hinzu.
2. Wählen Sie Neues Profil und geben Sie `emr-socks-proxy` als Profilnamen ein.
3. Wählen Sie PAC-Profil und dann Erstellen. Mithilfe von [PAC-Dateien \(Proxy Auto Configuration\)](#) können Sie eine Zulassungsliste für Browseranfragen definieren, die an einen Web-Proxyserver weitergeleitet werden sollen.
4. Ersetzen Sie im Feld PAC-Skript den Inhalt durch das folgende Skript, das definiert, was über Ihren Web-Proxyserver weitergeleitet werden URLs soll. Wenn Sie bei der Einrichtung Ihres

SSH-Tunnels eine andere Portnummer angegeben haben, **8157** ersetzen Sie diese durch Ihre Portnummer.

```
function FindProxyForURL(url, host) {  
    if (shExpMatch(url, "*ec2*.compute*.amazonaws.com*")) return 'SOCKS5  
    localhost:8157';  
    if (shExpMatch(url, "*ec2*.compute*")) return 'SOCKS5 localhost:8157';  
    if (shExpMatch(url, "http://10.*")) return 'SOCKS5 localhost:8157';  
    if (shExpMatch(url, "*10*.compute*")) return 'SOCKS5 localhost:8157';  
    if (shExpMatch(url, "*10*.amazonaws.com*")) return 'SOCKS5 localhost:8157';  
    if (shExpMatch(url, ".*.compute.internal*")) return 'SOCKS5 localhost:8157';  
    if (shExpMatch(url, "*ec2.internal*")) return 'SOCKS5 localhost:8157';  
    return 'DIRECT';  
}
```

5. Wählen Sie unter Aktionen die Option Änderungen übernehmen aus, um Ihre Proxyeinstellungen zu speichern.
6. Wählen Sie in der Chrome-Symbolleiste das `emr-socks-proxy` Profil aus SwitchyOmega und wählen Sie es aus.

Im Browser auf eine Weboberfläche zugreifen

Um eine Weboberfläche zu öffnen, geben Sie den öffentlichen DNS-Namen Ihres Primär- oder Core-Knotens gefolgt von der Portnummer für die von Ihnen gewählte Schnittstelle in die Adressleiste Ihres Browsers ein. Das folgende Beispiel zeigt die URL, die Sie eingeben würden, um eine Verbindung zum Spark herzustellen HistoryServer.

```
http://master-public-dns-name:18080/
```

Anweisungen zum Abrufen des öffentlichen DNS-Namens eines Knotens finden Sie unter [Abrufen des öffentlichen DNS-Namens für den Primärknoten](#). Eine vollständige Liste der URLs Webschnittstellen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Arbeit an einen Amazon EMR-Cluster einreichen

In diesem Abschnitt werden die Methoden beschrieben, mit denen Sie Arbeiten an einen Amazon-EMR-Cluster einreichen können. Um Arbeit einzureichen, können Sie Schritte hinzufügen oder interaktiv Hadoop-Jobs an den Primärknoten senden.

Beachten Sie beim Einreichen von Schritten an einen Cluster die folgenden Verhaltensregeln:

- Eine Schritt-ID kann bis zu 256 Zeichen enthalten.
- Sie können bis zu 256 PENDING- und RUNNING-Schritte in einem Cluster haben.
- Sie können Aufträge interaktiv an den Primärknoten übermitteln, auch dann, wenn auf dem Cluster 256 aktive Schritte ausgeführt werden. Sie können unbegrenzt viele Schritte während der Nutzungsdauer eines langlebigen Clusters übermitteln, es können aber nur 256 Schritte zu einem bestimmten Zeitpunkt den Status RUNNING oder PENDING haben.
- Mit Amazon-EMR-Versionen 4.8.0 und höher, außer Version 5.0.0, können Sie ausstehende Schritte abbrechen. Weitere Informationen finden Sie unter [Schritte abbrechen, wenn Sie Arbeit an einen Amazon EMR-Cluster einreichen](#).
- Mit Amazon-EMR-Version 5.28.0 und höher können Sie sowohl ausstehende als auch ausgeführte Schritte abbrechen. Sie können auch mehrere Schritte parallel ausführen, um die Clusterauslastung zu verbessern und Kosten zu sparen. Weitere Informationen finden Sie unter [Überlegungen zur parallel Ausführung mehrerer Schritte beim Einreichen von Arbeiten an Amazon EMR](#).

 Note

Für eine optimale Leistung empfehlen wir, benutzerdefinierte Bootstrap-Aktionen, -Skripts und andere Dateien, die Sie mit Amazon EMR verwenden möchten, in einem Amazon S3 S3-Bucket zu speichern, der sich in demselben Cluster AWS-Region befindet.

Themen

- [Hinzufügen von Schritten zu einem Cluster mit der Verwaltungskonsole für Amazon EMR](#)
- [Hinzufügen von Schritten zu einem Amazon EMR-Cluster mit dem AWS CLI](#)
- [Überlegungen zur parallel Ausführung mehrerer Schritte beim Einreichen von Arbeiten an Amazon EMR](#)
- [Schritte nach dem Einreichen von Arbeiten an einen Amazon EMR-Cluster anzeigen](#)
- [Schritte abbrechen, wenn Sie Arbeit an einen Amazon EMR-Cluster einreichen](#)

Hinzufügen von Schritten zu einem Cluster mit der Verwaltungskonsole für Amazon EMR

Verwenden Sie die folgenden Verfahren, um Schritte zu einem Cluster mit dem AWS Management Console hinzuzufügen. Detaillierte Informationen zum Einreichen von Schritten für bestimmte Big-Data-Anwendungen finden Sie in den folgenden Abschnitten des [Amazon-EMR-Versionshandbuchs](#):

- [Einen benutzerdefinierten JAR-Schritt senden](#)
- [Einen Hadoop-Streaming-Schritt senden](#)
- [Einen Spark-Schritt senden](#)
- [Einen Pig-Schritt senden](#)
- [Einen Befehl oder ein Skript als Schritt ausführen](#)
- [Werte in Schritte übergeben, um Hive-Skripte auszuführen](#)

So fügen Sie Schritte während der Clustererstellung hinzu

Von dort AWS Management Console aus können Sie beim Erstellen eines Clusters Schritte hinzufügen.

Console

Um Schritte hinzuzufügen, wenn Sie einen Cluster mit der Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie unter Schritte die Option Schritt hinzufügen aus. Geben Sie die entsprechenden Werte in die Felder im Dialogfeld Schritt hinzufügen ein. Informationen zur Formatierung Ihrer Schrittargumente finden Sie unter [Schritt-Argumente hinzufügen](#). Die Optionen unterscheiden sich je nach Schritttyp. Um Ihren Schritt hinzuzufügen und das Dialogfeld zu verlassen, wählen Sie Schritt hinzufügen.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

So fügen Sie einem ausgeführten Cluster Schritte hinzu

Mit dem können Sie einem Cluster Schritte hinzufügen AWS Management Console, bei denen die Option zum automatischen Beenden deaktiviert ist.

Console

Um einem laufenden Cluster mit der Konsole Schritte hinzuzufügen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Wählen Sie auf der Seite „Clusterdetails“ auf der Registerkarte Schritte die Option Schritt hinzufügen aus. Um einen vorhandenen Schritt zu klonen, wählen Sie das Dropdownmenü Aktionen und dann Schritt klonen aus.
4. Geben Sie die entsprechenden Werte in die Felder im Dialogfeld Schritt hinzufügen ein. Die Optionen unterscheiden sich je nach Schritttyp. Um Ihren Schritt hinzuzufügen und das Dialogfeld zu verlassen, wählen Sie Schritt hinzufügen.

So ändern Sie die Nebenläufigkeitsstufe für Schritte in einem ausgeführten Cluster

Mit dem AWS Management Console können Sie die Stufe der Schrittparallelität in einem laufenden Cluster ändern.

Note

Sie können mehrere Schritte nur mit Amazon EMR Version 5.28.0 und höher parallel ausführen.

Console

Um die Schrittparallelität in einem laufenden Cluster mit der Konsole zu ändern

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.

2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten. Der Cluster muss ausgeführt werden, um sein Parallelitätsattribut zu ändern.
3. Suchen Sie auf der Seite mit den Cluster-Details auf der Registerkarte Schritte den Abschnitt Attribute. Wählen Sie Bearbeiten aus, um die Parallelität zu ändern. Geben Sie einen Wert zwischen 1 und 256 ein.

Schritt-Argumente hinzufügen

Wenn Sie den verwenden AWS Management Console , um Ihrem Cluster einen Schritt hinzuzufügen, können Sie Argumente für diesen Schritt im Feld Argumente angeben. Sie müssen Argumente durch Leerzeichen trennen und Zeichenkettenargumente einschließen, die aus Zeichen und Leerzeichen in Anführungszeichen bestehen.

Example : Richtige Argumente

Die folgenden Beispielargumente sind für das korrekt formatiert AWS Management Console, wobei das letzte Zeichenkettenargument in Anführungszeichen gesetzt ist.

```
bash -c "aws s3 cp s3://amzn-s3-demo-bucket/my-script.sh ."
```

Sie können auch jedes Argument aus Gründen der besseren Lesbarkeit in eine separate Zeile einfügen, wie im folgenden Beispiel gezeigt.

```
bash
-c
"aws s3 cp s3://amzn-s3-demo-bucket/my-script.sh ."
```

Example : Falsche Argumente

Die folgenden Beispielargumente sind falsch formatiert für AWS Management Console. Beachten Sie, dass das letzte Zeichenkettenargument, `aws s3 cp s3://amzn-s3-demo-bucket/my-script.sh .`, Leerzeichen enthält und nicht von Anführungszeichen umgeben ist.

```
bash -c aws s3 cp s3://amzn-s3-demo-bucket/my-script.sh .
```

Hinzufügen von Schritten zu einem Amazon EMR-Cluster mit dem AWS CLI

Die folgenden Verfahren zeigen, wie Sie Schritte zu einem neu erstellten Cluster und zu einem aktiven Cluster mit der AWS CLI hinzufügen. In beiden Beispielen wird der Unterbefehl `--steps` verwendet, um Schritte zum Cluster hinzuzufügen.

So fügen Sie Schritte während der Clustererstellung hinzu

- Geben Sie den folgenden Befehl ein, um einen Cluster zu erstellen und einen Apache Pig-Schritt hinzuzufügen. Stellen Sie sicher, dass Sie es *myKey* durch den Namen Ihres EC2 Amazon-Schlüsselpaars ersetzen.

```
aws emr create-cluster --name "Test cluster" \
--applications Name=Spark \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-groups InstanceGroupType=PRIMARY,InstanceCount=1,InstanceType=m5.xlarge \
InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-runner.jar","Properties":"","Name":"Spark application"}]'
```

Note

Die Liste der Argumente ändert sich je nach Art des Schritts.

Standardmäßig ist Nebenläufigkeitsstufe für Schritte 1. Sie können die Nebenläufigkeitsstufe für Schritte festlegen, indem Sie den `StepConcurrencyLevel`-Parameter beim Erstellen eines Clusters verwenden.

Die Ausgabe ist eine Cluster-Kennung ähnlich der folgenden.

```
{
  "ClusterId": "j-2AXXXXXXGAPLF"
}
```

So fügen Sie einen Schritt einem aktiven Cluster hinzu

- Geben Sie den folgenden Befehl ein, um einen Schritt zu einem aktiven Cluster hinzuzufügen. Ersetzen Sie `j-2AXXXXXXGAPLF` durch die ID Ihres eigenen Clusters.

```
aws emr add-steps --cluster-id j-2AXXXXXXGAPLF \  
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--  
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-  
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-  
runner.jar","Properties":"","Name":"Spark application"}]'
```

Die Ausgabe ist eine Schrittkennung ähnlich der folgenden.

```
{  
  "StepIds": [  
    "s-Y9XXXXXXAPMD"  
  ]  
}
```

Um das StepConcurrencyLevel in einem laufenden Cluster zu ändern

- In einem laufenden Cluster können Sie den `StepConcurrencyLevel` mit der `ModifyCluster-API` ändern. Geben Sie beispielsweise den folgenden Befehl ein, um die `StepConcurrencyLevel` für Schritte auf 10 zu erhöhen. Ersetzen Sie `j-2AXXXXXXGAPLF` durch die ID Ihres Clusters.

```
aws emr modify-cluster --cluster-id j-2AXXXXXXGAPLF --step-concurrency-level 10
```

- Die Ausgabe sieht folgendermaßen oder ähnlich aus.

```
{  
  "StepConcurrencyLevel": 10  
}
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen in der AWS CLI finden Sie in der [AWS CLI Befehlsreferenz](#).

Überlegungen zur parallel Ausführung mehrerer Schritte beim Einreichen von Arbeiten an Amazon EMR

Die parallel Ausführung mehrerer Schritte bei der Einreichung von Arbeiten an Amazon EMR erfordert vorläufige Entscheidungen über die Ressourcenplanung und die Erwartungen in Bezug auf das Clusterverhalten. Diese werden hier ausführlich behandelt.

- Parallel laufende Schritte können in beliebiger Reihenfolge abgeschlossen werden, aber ausstehende Schritte in der Warteschlange gehen in der Reihenfolge in den laufenden Zustand über, in der sie eingereicht wurden.
- Wenn Sie eine Nebenläufigkeitsstufe für Schritte für den Cluster auswählen, müssen Sie überlegen, ob der Primärknoten-Instance-Typ die Speicheranforderungen von Benutzer-Workloads erfüllt. Der Hauptschrittausführungsprozess wird für jeden Schritt auf dem Primärknoten ausgeführt. Das parallele Ausführen mehrerer Schritte erfordert mehr Arbeitsspeicher und eine höhere CPU-Auslastung auf dem Primärknoten als die Ausführung eines einzelnen Schrittes.
- Um eine komplexe Planung und Ressourcenverwaltung von gleichzeitigen Schritten zu erreichen, können Sie YARN-Planungsfunktionen wie `FairScheduler` oder verwenden `CapacityScheduler`. Beispielsweise können Sie `FairScheduler` mit einem `queueMaxAppsDefault`-Satz verwenden, um zu verhindern, dass mehr als eine bestimmte Anzahl von Aufgaben gleichzeitig ausgeführt werden.
- Die Nebenläufigkeitsstufe für Schritte unterliegt den Konfigurationen von Ressourcenmanagern. Wenn YARN beispielsweise nur mit einer Parallelität von 5 konfiguriert ist, können Sie nur fünf YARN-Anwendungen parallel laufen lassen, selbst wenn `StepConcurrencyLevel` auf 10 gesetzt ist. Weitere Informationen finden Sie unter [Konfigurieren von Anwendungen](#) in den Amazon-EMR-Versionshinweisen.
- Sie können keinen Schritt mit einem anderen `ActionOnFailure` als `CONTINUE` hinzufügen, solange die Schrittparallelitätsstufe des Clusters größer als 1 ist.
- Wenn die Step-Parallelitätsstufe eines Clusters größer als eins ist, wird das `ActionOnFailure`-Step-Feature nicht aktiviert.
- Wenn ein Cluster über die Schritt-Parallelitätsstufe 1, aber über mehrere laufende Schritte verfügt, wird `TERMINATE_CLUSTER` `ActionOnFailure` möglicherweise aktiviert, `CANCEL_AND_WAIT` `ActionOnFailure` jedoch nicht. Dieser Grenzfall tritt auf, wenn die Parallelitätsstufe für Clusterschritte höher als eins war, aber während der Ausführung mehrerer Schritte niedriger war.
- Sie können EMR Auto Scaling verwenden, um basierend auf den YARN-Ressourcen vertikal zu skalieren und Ressourcenkonflikte zu vermeiden. Weitere Informationen finden Sie unter

[Verwenden der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instance-Gruppen](#) im Verwaltungshandbuch für Amazon EMR.

- Wenn Sie die Nebenläufigkeitsstufe für Schritte verringern, erlaubt EMR das Abschließen aller laufenden Schritte, bevor die Anzahl der Schritte reduziert wird. Wenn die Ressourcen ausgeschöpft sind, weil der Cluster zu viele gleichzeitige Schritte ausführt, empfehlen wir, alle laufenden Schritte manuell abubrechen, um Ressourcen freizumachen.

Schritte nach dem Einreichen von Arbeiten an einen Amazon EMR-Cluster anzeigen

Sie können bis zu 10.000 Schritte sehen, die Amazon EMR in den letzten sieben Tagen abgeschlossen hat. Sie können sich auch jederzeit 1.000 Schritte ansehen, die Amazon EMR abgeschlossen hat. Diese Gesamtzahl umfasst sowohl vom Benutzer übermittelte Schritte als auch Systemschritte.

Wenn Sie neue Schritte einreichen, sobald der Cluster das Datensatzlimit von 1.000 Schritten erreicht hat, löscht Amazon EMR die inaktiven, von Benutzern eingereichten Schritte, deren Status seit mehr als sieben Tagen ABGESCHLOSSEN, STORNIERT oder FEHLGESCHLAGEN lautet. Wenn Sie Schritte einreichen, die das Limit von 10.000 Schrittdatensätzen überschreiten, löscht Amazon EMR die inaktiven, von Benutzern eingereichten Schrittdatensätze unabhängig von ihrer inaktiven Dauer. Amazon EMR entfernt diese Datensätze nicht aus den Protokolldateien. Amazon EMR entfernt sie aus der AWS Konsole und sie werden nicht zurückgegeben, wenn Sie die API AWS CLI oder verwenden, um Cluster-Informationen abzurufen. Systemschrittdatensätze werden niemals entfernt.

Welche Schrittinformationen Sie anzeigen können, hängt vom Mechanismus ab, der zum Abrufen der Cluster-Informationen verwendet wurde. Die folgende Tabelle enthält die Schrittinformationen, die von den verfügbaren Optionen jeweils zurückgegeben werden.

Option	DescribeJobFlow oder --describe --jobflow	ListSteps oder liste-steps
SDK	256 Schritte	Bis zu 10.000 Schritte
Amazon-EMR-CLI	256 Schritte	N/A
AWS CLI	N/A	Bis zu 10.000 Schritte

Option	DescribeJobFlow oder --describe --jobflow	ListSteps oder liste-steps
API	256 Schritte	Bis zu 10.000 Schritte

Schritte abbrechen, wenn Sie Arbeit an einen Amazon EMR-Cluster einreichen

Sie können ausstehende und laufende Schritte aus dem AWS Management Console AWS CLI, dem oder dem Amazon EMR stornieren, wenn Sie Arbeit an Ihren Cluster einreichen. API.

Console

Um Schritte mit der Konsole abzuberechnen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters aus, und wählen Sie dann den Cluster aus, den Sie aktualisieren möchten.
3. Klicken Sie auf der Seite Cluster-Details auf der Registerkarte Schritte auf das Kontrollkästchen neben dem Schritt, den Sie abbrechen möchten. Wählen Sie das Dropdownmenü Aktionen und dann Schritte abbrechen aus.
4. Wählen Sie im Dialogfeld Schritt abbrechen entweder den Schritt abbrechen und warten, bis er beendet ist, oder ob Sie den Schritt abbrechen und das Beenden erzwingen möchten. Wählen Sie dann Confirm (Bestätigen) aus.
5. Der Status der Schritte in der Tabelle Schritte ändert sich in CANCELLED.

CLI

Zum Stornieren verwenden Sie AWS CLI

- Verwenden Sie den Befehl `aws emr cancel-steps` unter Angabe des Clusters und der abzuberechnenden Schritte. Das folgende Beispiel zeigt einen AWS CLI -Befehl für den Abbruch von zwei Schritten.

```
aws emr cancel-steps --cluster-id j-2QUAXXXXXXXXXX \
```



```
--step-ids s-3M8DXXXXXXXXXX s-3M8DXXXXXXXXXX \  
--step-cancellation-option SEND_INTERRUPT
```

Mit Amazon-EMR-Version 5.28.0 können Sie beim Abbrechen von Schritten eine der beiden folgenden Abbruchoptionen für StepCancellationOption-Parameter auswählen.

- **SEND_INTERRUPT** – Dies ist die Standardoption. Wenn eine Anfrage zum Abbruch eines Schritts eingeht, sendet EMR ein SIGTERM-Signal an den Schritt. Fügen Sie Ihrer Schrittlogik einen SIGTERM-Signal-Handler hinzu, um dieses Signal abzufangen und die Prozesse der untergeordneten Schritte zu beenden oder zu warten, bis sie abgeschlossen sind.
- **TERMINATE_PROCESS** – Wenn diese Option ausgewählt ist, sendet EMR ein SIGKILL-Signal an den Schritt und alle seine untergeordneten Prozesse, wodurch sie sofort beendet werden.

Was es bei der Stornierung von Schritten zu berücksichtigen gibt

- Wenn Sie einen laufenden oder ausstehenden Schritt abbrechen, wird dieser Schritt aus der aktiven Schrittzahl entfernt.
- Wenn Sie einen laufenden Schritt abbrechen, kann ein ausstehender Schritt nicht ausgeführt werden, vorausgesetzt, dass keine Änderung an stepConcurrencyLevel vorgenommen wurde.
- Durch das Abbrechen eines laufenden Schritts wird der Schritt ActionOnFailure nicht ausgelöst.
- **SEND_INTERRUPT** StepCancellationOption sendet für EMR 5.32.0 und höher ein SIGTERM-Signal an den untergeordneten Schrittprozess. Sie sollten auf dieses Signal achten und eine Säuberung durchführen und das System ordnungsgemäß herunterfahren. **TERMINATE_PROCESS** StepCancellationOption sendet ein SIGKILL-Signal an den untergeordneten Schrittprozess und alle seine untergeordneten Prozesse. Asynchrone Prozesse sind jedoch nicht betroffen.

Einen Amazon EMR-Cluster bei der Ausführung seiner Arbeit anzeigen und überwachen

Amazon EMR bietet mehrere Tools, die Sie verwenden können, um Informationen über Ihren Cluster zu sammeln. Sie können Informationen zum Cluster über die Konsole, die Befehlszeilenschnittstelle (CLI) oder programmgesteuert abrufen. Die Standard-Hadoop-Webschnittstellen und Protokolldateien sind auf dem Primärknoten verfügbar. Sie können auch Überwachungsdienste wie CloudWatch Ganglia verwenden, um die Leistung Ihres Clusters zu verfolgen.

Der Anwendungsverlauf ist ab Amazon EMR 5.25.0 auch über die Konsole mit der „persistenten“ Anwendung UIs für Spark History Server verfügbar. Mit Amazon EMR 6.x sind auch persistente YARN Timeline Server und Tez-Benutzeroberflächen verfügbar. Diese Dienste werden außerhalb des Clusters gehostet, sodass Sie nach Beendigung des Clusters 30 Tage lang auf den Anwendungsverlauf zugreifen können, ohne dass eine SSH-Verbindung oder ein Web-Proxy erforderlich ist. Weitere Information unter [Anwendungsverlauf anzeigen](#)

Themen

- [Status und Details des Amazon EMR-Clusters anzeigen](#)
- [Verbessertes schrittweises Debugging mit Amazon EMR](#)
- [Amazon EMR-Anwendungsverlauf anzeigen](#)
- [Amazon EMR-Protokolldateien anzeigen](#)
- [Cluster-Instances in Amazon anzeigen EC2](#)
- [CloudWatch Ereignisse und Metriken von Amazon EMR](#)
- [Cluster-Anwendungsmetriken mithilfe von Ganglia mit Amazon EMR anzeigen](#)
- [Protokollieren AWS EMR EMR-API-Aufrufen mit AWS CloudTrail](#)
- [Bewährte Methoden zur EMR-Observabilität](#)

Status und Details des Amazon EMR-Clusters anzeigen

Nach dem Erstellen eines Clusters können Sie seinen Status überwachen und detaillierte Informationen zu seiner Ausführung sowie eventuell aufgetretenen Fehlern erhalten – auch nach dem Beenden des Clusters. Amazon EMR speichert für Ihre Referenz Metadaten über beendete Cluster für zwei Monate. Anschließend werden die Metadaten gelöscht. Sie können keine Cluster aus dem Cluster-Verlauf löschen. Sie können jedoch in der AWS Management Console die Option Filter (Filtern) und in der AWS CLI Optionen mit dem Befehl `list-clusters` verwenden, um sich auf für Sie relevante Cluster zu konzentrieren.

Sie können auf den innerhalb des Clusters gespeicherten Anwendungsverlauf eine Woche ab dem Zeitpunkt der Aufzeichnung zugreifen, unabhängig davon, ob der Cluster ausgeführt wird oder beendet wurde. Darüber hinaus speichern persistente Anwendungsbenutzeroberflächen den Anwendungsverlauf für 30 Tage nach Beendigung eines Clusters außerhalb des Clusters. Weitere Information unter [Anwendungsverlauf anzeigen](#)

Weitere Hinweise zu Clusterstatus, wie Wartend und Läuft, finden Sie unter [Verstehen des Cluster-Lebenszyklus](#).

So lassen Sie Cluster-Details mithilfe der AWS Management Console anzeigen

In der Clusterliste unter <https://console.aws.amazon.com/emr> sind alle Cluster in Ihrem Konto und Ihrer AWS Region aufgeführt, einschließlich beendeter Cluster. Die Liste enthält für jeden Cluster Folgendes: den Namen und die ID, die Status- und Statusdetails, die Erstellungszeit, die abgelaufene Zeit, in der der Cluster ausgeführt wurde, und die normalisierten Instanzstunden, die für alle Instances im Cluster angefallen sind. EC2 Diese Liste ist der Ausgangspunkt für die Überwachung des Status von Clustern. Sie ist so konzipiert, dass Sie jeden Cluster zu Analyse- und Fehlerbehebungs Zwecken aufschlüsseln können.

Console

Um Clusterinformationen mit der Konsole anzuzeigen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie anzeigen möchten.
3. Im Bereich Zusammenfassung können Sie sich die Grundlagen Ihrer Cluster-Konfiguration ansehen, z. B. den Cluster-Status, die Open-Source-Anwendungen, die Amazon EMR auf dem Cluster installiert hat, und die Version von Amazon EMR, mit der Sie den Cluster erstellt haben. Verwenden Sie jede Registerkarte unterhalb der Zusammenfassung, um die in der folgenden Tabelle beschriebenen Informationen anzuzeigen.

Zeigen Sie Clusterdetails an, indem Sie AWS CLI

Die folgenden Beispiele zeigen, wie Sie Cluster-Details über die AWS CLI abrufen. Weitere Informationen zu verfügbaren Befehlen finden Sie in der [AWS CLI -Befehlsreferenz für Amazon EMR](#). Sie können den Befehl [describe-cluster](#) verwenden, um Cluster-Details einschließlich Status, Hardware- und Softwarekonfiguration, VPC-Einstellungen, Bootstrap-Aktionen, Instance-Gruppen usw. anzuzeigen. Weitere Informationen zu Cluster-Status finden Sie unter [Verstehen des Cluster-Lebenszyklus](#). Das folgende Beispiel zeigt die Verwendung des Befehls `describe-cluster`, gefolgt von Beispielen für den Befehl [list-clusters](#).

Example Anzeigen des Cluster-Status

Sie benötigen die Cluster-ID, um den Befehl `describe-cluster` zu verwenden. Dieses Beispiel zeigt, wie Sie eine Liste von Clustern abrufen, die innerhalb eines bestimmten Datumsbereichs

erstellt wurden, und anschließend mithilfe eines der IDs zurückgegebenen Cluster weitere Informationen zum Status eines einzelnen Clusters auflisten können.

Der folgende Befehl beschreibt den Cluster `j-1K48XXXXXXHCB`, den Sie durch Ihre Cluster-ID ersetzen.

```
aws emr describe-cluster --cluster-id j-1K48XXXXXXHCB
```

Die Ausgabe des Befehls ähnelt der folgenden:

```
{
  "Cluster": {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1438281058.061,
        "CreationDateTime": 1438280702.498
      },
      "State": "WAITING",
      "StateChangeReason": {
        "Message": "Waiting for steps to run"
      }
    },
    "Ec2InstanceAttributes": {
      "EmrManagedMasterSecurityGroup": "sg-cXXXXX0",
      "IamInstanceProfile": "EMR_EC2_DefaultRole",
      "Ec2KeyName": "myKey",
      "Ec2AvailabilityZone": "us-east-1c",
      "EmrManagedSlaveSecurityGroup": "sg-example"
    },
    "Name": "Development Cluster",
    "ServiceRole": "EMR_DefaultRole",
    "Tags": [],
    "TerminationProtected": false,
    "ReleaseLabel": "emr-4.0.0",
    "NormalizedInstanceHours": 16,
    "InstanceGroups": [
      {
        "RequestedInstanceCount": 1,
        "Status": {
          "Timeline": {
            "ReadyDateTime": 1438281058.101,
            "CreationDateTime": 1438280702.499
          }
        }
      }
    ]
  }
}
```

```

        "State": "RUNNING",
        "StateChangeReason": {
            "Message": ""
        }
    },
    "Name": "CORE",
    "InstanceGroupType": "CORE",
    "Id": "ig-2EEXAMPLEXP",
    "Configurations": [],
    "InstanceType": "m5.xlarge",
    "Market": "ON_DEMAND",
    "RunningInstanceCount": 1
},
{
    "RequestedInstanceCount": 1,
    "Status": {
        "Timeline": {
            "ReadyDateTime": 1438281023.879,
            "CreationDateTime": 1438280702.499
        },
        "State": "RUNNING",
        "StateChangeReason": {
            "Message": ""
        }
    },
    "Name": "MASTER",
    "InstanceGroupType": "MASTER",
    "Id": "ig-2A1234567XP",
    "Configurations": [],
    "InstanceType": "m5.xlarge",
    "Market": "ON_DEMAND",
    "RunningInstanceCount": 1
}
],
"Applications": [
    {
        "Version": "1.0.0",
        "Name": "Hive"
    },
    {
        "Version": "2.6.0",
        "Name": "Hadoop"
    }
]

```

```

        "Version": "0.14.0",
        "Name": "Pig"
    },
    {
        "Version": "1.4.1",
        "Name": "Spark"
    }
],
"BootstrapActions": [],
"MasterPublicDnsName": "ec2-X-X-X-X.compute-1.amazonaws.com",
"AutoTerminate": false,
"Id": "j-jobFlowID",
"Configurations": [
    {
        "Properties": {
            "hadoop.security.groups.cache.secs": "250"
        },
        "Classification": "core-site"
    },
    {
        "Properties": {
            "mapreduce.tasktracker.reduce.tasks.maximum": "5",
            "mapred.tasktracker.map.tasks.maximum": "2",
            "mapreduce.map.sort.spill.percent": "90"
        },
        "Classification": "mapred-site"
    },
    {
        "Properties": {
            "hive.join.emit.interval": "1000",
            "hive.merge.mapfiles": "true"
        },
        "Classification": "hive-site"
    }
]
}

```

Example Auflisten von Clustern nach Erstellungsdatum

Zum Abrufen von in einem bestimmten Datenbereich erstellten Clustern verwenden Sie den Befehl `list-clusters` mit den Parametern `--created-after` und `--created-before`.

Mit dem folgenden Befehl werden alle Cluster aufgelistet, die zwischen dem 9. Oktober 2019 und dem 12. Oktober 2019 erstellt wurden.

```
aws emr list-clusters --created-after 2019-10-09T00:12:00 --created-before 2019-10-12T00:12:00
```

Example Auflisten von Clustern nach Status

Verwenden Sie zum Auflisten von Clustern nach Status den Befehl `list-clusters` mit dem Parameter `--cluster-states`. Zu den zulässigen Cluster-Status gehören: `STARTING`, `BOOTSTRAPPING`, `RUNNING`, `WAITING`, `TERMINATING`, `TERMINATED` und `TERMINATED_WITH_ERRORS`.

```
aws emr list-clusters --cluster-states TERMINATED
```

Sie können auch die folgenden Abkürzungsparameter verwenden, um alle Cluster in den angegebenen Zuständen aufzulisten:

- `--active` filtert Cluster mit den Status `STARTING`, `BOOTSTRAPPING`, `RUNNING`, `WAITING` oder `TERMINATING`.
- `--terminated` filtert Cluster mit dem Status `TERMINATED`.
- `--failed` filtert Cluster mit dem Status `TERMINATED_WITH_ERRORS`.

Die folgenden Befehle geben dasselbe Ergebnis zurück.

```
aws emr list-clusters --cluster-states TERMINATED
```

```
aws emr list-clusters --terminated
```

Weitere Informationen zu Cluster-Status finden Sie unter [Verstehen des Cluster-Lebenszyklus](#).

Verbessertes schrittweises Debugging mit Amazon EMR

Wenn ein Amazon-EMR-Schritt fehlschlägt und Sie Ihre Arbeit mit der Schritt-API-Operation mit einem AMI, Version 5.x oder höher, gesendet haben, kann Amazon EMR in einigen Fällen die Ursache des Schrittfehlers ermitteln und zusammen mit den Namen der entsprechenden

Protokolldatei und einem Teil der Anwendungs-Stack-Trace-Informationen über die API zurückgeben. Die folgenden Fehler können identifiziert werden:

- Ein üblicher Hadoop-Fehler, wie z. B. das Ausgabeverzeichnis ist bereits vorhanden, das Eingabeverzeichnis ist nicht vorhanden oder für eine Anwendung ist nicht mehr genügend Speicherplatz vorhanden.
- Java-Fehler, wie z. B. eine Anwendung, die mit einer inkompatiblen Version von Java kompiliert und mit einer Hauptklasse ausgeführt wurde, die nicht gefunden wird.
- Ein Problem mit dem Zugriff auf Objekte, die in Amazon S3 gespeichert sind.

Diese Informationen sind mithilfe der [ListSteps](#) API-Operationen [DescribeStep](#) und [FailureDetails](#) verfügbar. Das [FailureDetails](#) Feld der von diesen Operationen [StepSummary](#) zurückgegebenen. Verwenden Sie die AWS CLI, die Konsole oder das AWS SDK, um auf die FailureDetails Informationen zuzugreifen.

Console

Die neue Amazon-EMR-Konsole bietet kein schrittweises Debugging. Mit den folgenden Schritten können Sie jedoch Details zur Clusterbeendigung anzeigen.

Um Fehlerdetails mit der Konsole anzuzeigen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters aus, und wählen Sie dann den Cluster aus, den Sie anzeigen möchten.
3. Notieren Sie sich den Statuswert im Abschnitt Zusammenfassung der Cluster-Detailseite. Wenn der Status Mit Fehlern beendet lautet, bewegen Sie den Mauszeiger über den Text, um Details zum Clusterausfall anzuzeigen.

CLI

Um Fehlerdetails anzuzeigen, verwenden Sie AWS CLI

- Verwenden Sie den `describe-step` Befehl AWS CLI, um Fehlerdetails für einen Schritt mit dem abzurufen.

```
aws emr describe-step --cluster-id j-1K48XXXXXHCB --step-id s-3QM0XXXXXM1W
```


Die Ausgabe sieht etwa folgendermaßen aus:

```
{
  "Step": {
    "Status": {
      "FailureDetails": {
        "LogFile": "s3://amzn-s3-demo-bucket/logs/j-1K48XXXXXHCB/steps/
s-3QM0XXXXXXM1W/stderr.gz",
        "Message": "org.apache.hadoop.mapred.FileAlreadyExistsException: Output
directory s3://amzn-s3-demo-bucket/logs/beta already exists",
        "Reason": "Output directory already exists."
      },
      "Timeline": {
        "EndDateTime": 1469034209.143,
        "CreationDateTime": 1469033847.105,
        "StartDateTime": 1469034202.881
      },
      "State": "FAILED",
      "StateChangeReason": {}
    },
    "Config": {
      "Args": [
        "wordcount",
        "s3://amzn-s3-demo-bucket/input/input.txt",
        "s3://amzn-s3-demo-bucket/logs/beta"
      ],
      "Jar": "s3://amzn-s3-demo-bucket/jars/hadoop-mapreduce-examples-2.7.2-
amzn-1.jar",
      "Properties": {}
    },
    "Id": "s-3QM0XXXXXXM1W",
    "ActionOnFailure": "CONTINUE",
    "Name": "ExampleJob"
  }
}
```

Amazon EMR-Anwendungsverlauf anzeigen

Sie können die Anwendungsdetails für den Spark History Server und den YARN-Timeline-Dienst auf der Detailseite des Clusters in der Konsole einsehen. Durch die Verwendung des Amazon-EMR-

Anwendungsverlaufs wird es einfacher, Probleme mit aktiven Aufträgen und dem Auftragsverlauf zu beheben und zu analysieren.

 Note

Um die Sicherheit der Anwendungen außerhalb der Konsole zu erhöhen, die Sie möglicherweise mit Amazon EMR verwenden, sind die Anwendungs-Hosting-Domains in der Public Suffix List (PSL) registriert. Zu diesen Hosting-Domains gehören beispielsweise die folgenden: `emrstudio-prod.us-east-1.amazonaws.com`, `emrnotebooks-prod.us-east-1.amazonaws.com`, `emrappui-prod.us-east-1.amazonaws.com`. Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host--`-Präfix zu verwenden, falls Sie jemals sensible Cookies im Standard-Domainnamen einrichten müssen. Diese Vorgehensweise hilft Ihnen dabei, Ihre Domain vor CSRF (Cross-Site Request Forgery Attempts, Anforderungsfälschung zwischen Websites)-Versuchen zu schützen. Weitere Informationen finden Sie auf der [Set-Cookie](#)-Seite im Mozilla Developer Network.

Der Abschnitt Anwendungsbenutzeroberflächen auf der Registerkarte Anwendungen bietet verschiedene Anzeigeeoptionen, abhängig vom Clusterstatus und den Anwendungen, die Sie auf dem Cluster installiert haben.

- [Off-Cluster-Zugriff auf persistente Anwendungsbenutzeroberflächen](#) – Ab Amazon-EMR-Version 5.25.0 sind persistente Anwendungsbenutzeroberflächenlinks für die Spark-Benutzeroberfläche und den Spark History Service verfügbar. Mit Amazon-EMR-Version 5.30.1 und höher verfügen Tez UI und der YARN Timeline Server auch über persistente Anwendungsbenutzeroberflächen. Der YARN Timeline Server und die Tez UI sind Open-Source-Anwendungen, die Metriken für aktive und beendete Cluster bereitstellen. Die Benutzeroberfläche von Spark enthält Details zu Planungsphasen und -aufgaben, RDD-Größen und Speichernutzung, Umgebungsinformationen und Informationen zu den laufenden Executoren. Dauerhafte Anwendungen UIs werden außerhalb des Clusters ausgeführt, sodass Clusterinformationen und -protokolle nach Beendigung einer Anwendung noch 30 Tage lang verfügbar sind. Im Gegensatz zu Benutzeroberflächen für Cluster-Anwendungen müssen Sie für persistente Anwendungen UIs keinen Webproxy über eine SSH-Verbindung einrichten.
- [Anwendungsbenutzeroberflächen innerhalb des Clusters](#) – Es gibt eine Vielzahl von Anwendungsverlauf-Benutzeroberflächen, die auf einem Cluster ausgeführt werden können. Benutzeroberflächen innerhalb eines Clusters werden auf dem Master-Knoten gehostet und erfordern, dass Sie eine SSH-Verbindung zum Webserver einrichten.

Anwendungsbenutzeroberflächen innerhalb eines Clusters speichern den Anwendungsverlauf für eine Woche nach dem Beenden einer Anwendung. Weitere Informationen und Anweisungen zum Einrichten eines SSH-Tunnels finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

Mit Ausnahme von Spark History Server-, YARN Timeline Server- und Hive-Anwendungen kann der Anwendungsverlauf innerhalb eines Clusters nur angezeigt werden, während der Cluster ausgeführt wird.

Persistente Anwendungsbenutzeroberflächen in Amazon EMR anzeigen

Ab Amazon EMR Version 5.25.0 können Sie eine Verbindung zu den persistenten Spark History Server-Anwendungsdetails herstellen, die außerhalb des Clusters gehostet werden, indem Sie die Seite Zusammenfassung des Clusters oder die Registerkarte Anwendungsbenutzeroberflächen in der Konsole verwenden. Persistente Anwendungsoberflächen für Tez UI und YARN Timeline Server sind ab Amazon-EMR-Version 5.30.1 verfügbar. Der Zugriff auf den persistenten Anwendungsverlauf mit einem Klick bietet folgende Vorteile:

- Sie können aktive Aufgaben und den Aufgabenverlauf schnell analysieren und Probleme damit beheben, ohne einen Web-Proxy über eine SSH-Verbindung einzurichten.
- Sie können auf den Anwendungsverlauf und relevante Protokolldateien für aktive und beendete Cluster zugreifen. Die Protokolle stehen nach dem Ende der Anwendung 30 Tage lang zur Verfügung.

Navigieren Sie in der Konsole zu Ihren Clusterdetails und wählen Sie die Registerkarte Applications (Anwendungen) aus. Wählen Sie die Benutzeroberfläche der Anwendung aus, die Sie nach dem Start Ihres Clusters verwenden möchten. Die Benutzeroberfläche der Anwendung wird in einer neuen Browserregisterkarte geöffnet. Weitere Informationen finden Sie unter [Überwachung und Instrumentierung](#).

Sie können YARN-Containerprotokolle über die Links auf dem Spark History Server, YARN Timeline Server und Tez UI anzeigen.

Note

Um über Spark History Servers, YARN Timeline Server und Tez UI auf YARN-Container-Protokolle zugreifen zu können, müssen Sie die Protokollierung in Amazon S3 für Ihren

Cluster aktivieren. Wenn Sie die Protokollierung nicht aktivieren, funktionieren die Links zu den YARN-Container-Protokollen nicht.

Protokollsammlung

Um den Zugriff auf persistente Anwendungsbenutzeroberflächen mit einem Klick zu ermöglichen, werden von Amazon EMR zwei Arten von Protokollen gesammelt:

- Anwendungsereignisprotokolle werden in einem EMR-System-Bucket erfasst. Die Ereignisprotokolle werden im Ruhezustand mittels serverseitiger Verschlüsselung mit Amazon S3 Managed Keys (SSE-S3) verschlüsselt. Wenn Sie ein privates Subnetz für Ihren Cluster verwenden, stellen Sie sicher, dass Sie den richtigen System-Bucket ARNs in die Ressourcenliste der Amazon S3 S3-Richtlinie für das private Subnetz aufnehmen. Weitere Informationen finden Sie unter [Amazon-S3-Mindestrichtlinie für privates Subnetz](#).
- YARN-Container-Protokolle werden in einem Amazon-S3-Bucket gesammelt, den Sie besitzen. Sie müssen die Protokollierung für Ihren Cluster aktivieren, um auf YARN-Container-Protokolle zugreifen zu können. Weitere Informationen finden Sie unter [Konfigurieren der Cluster-Protokollierung und des Debuggings](#).

Wenn Sie diese Funktion aus Datenschutzgründen deaktivieren müssen, können Sie den Daemon mithilfe eines Bootstrap-Skripts beim Erstellen eines Clusters stoppen, wie im folgenden Beispiel gezeigt wird.

```
aws emr create-cluster --name "Stop Application UI Support" --release-label emr-7.9.0 \
--applications Name=Hadoop Name=Spark --ec2-attributes KeyName=<myEMRKeyPairName> \
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m3.xlarge \
InstanceGroupType=CORE,InstanceCount=1,InstanceType=m3.xlarge \
InstanceGroupType=TASK,InstanceCount=1,InstanceType=m3.xlarge \
--use-default-roles --bootstrap-actions Path=s3://<region>.elasticmapreduce/bootstrap-
actions/run-if,Args=["instance.isMaster=true","echo Stop Application UI | sudo tee /
etc/apppusher/run-apppusher; sudo systemctl stop apppusher || exit 0"]
```

Nachdem Sie dieses Bootstrap-Skript ausgeführt haben, sammelt Amazon EMR keine Spark-History-Server- oder YARN-Timeline-Server-Ereignisprotokolle im EMR-System-Bucket. Auf der Registerkarte Application user interfaces (Anwendungsbenutzeroberflächen) werden keine Informationen zum Anwendungsverlauf verfügbar sein und Sie verlieren den Zugriff auf alle Anwendungsbenutzeroberflächen über die Konsole.

Große Spark-Ereignisprotokolldateien

In einigen Fällen können Spark-Jobs mit langer Laufzeit, wie Spark-Streaming, und große Jobs, wie Spark-SQL-Abfragen, große Ereignisprotokolle generieren. Bei umfangreichen Ereignisprotokollen können Sie schnell Festplattenspeicher auf Compute-Instances belegen und beim Laden von Persistent auf OutOfMemory Fehler stoßen. Um diese Probleme zu vermeiden, wird empfohlen, dass Sie das Feature zum Rollen und Verdichten von Spark-Ereignisprotokollen aktivieren. Dieses Feature ist nur in Amazon EMR ab Version emr-6.1.0 oder höher verfügbar. Weitere Informationen zum Rollen und Verdichten finden Sie in der Spark-Dokumentation unter [Anwenden der Komprimierung auf Protokolldateien für rollende Ereignisse](#).

Um das Feature zum Rollen und Verdichten des Spark-Ereignisprotokolls zu aktivieren, aktivieren Sie die folgenden Spark-Konfigurationseinstellungen.

- `spark.eventLog.rolling.enabled` – Aktiviert das Rolling des Ereignisprotokolls je nach Größe. Diese Einstellung ist standardmäßig deaktiviert.
- `spark.eventLog.rolling.maxFileSize` – Wenn das Rolling aktiviert ist, gibt dies die maximale Größe der Ereignisprotokolldatei an, bevor ein Rollover ausgeführt wird. Der Standardwert ist 128 MB.
- `spark.history.fs.eventLog.rolling.maxFilesToRetain` – Gibt die maximale Anzahl nicht komprimierter Ereignisprotokolldateien an, die aufbewahrt werden sollen. Standardmäßig werden alle Ereignisprotokolldateien aufbewahrt. Stellen Sie einen niedrigeren Wert ein, um ältere Ereignisprotokolle zu komprimieren. Der niedrigste Wert ist 1.

Beachten Sie, dass bei der Komprimierung versucht wird, Ereignisse mit veralteten Ereignisprotokolldateien auszuschließen, wie z. B. die folgenden. Wenn dabei Ereignisse verworfen werden, werden sie nicht mehr auf der Benutzeroberfläche des Spark History Servers angezeigt.

- Ereignisse für abgeschlossene Aufträge und zugehörige Phasen- oder Aufgabenereignisse.
- Ereignisse für beendete Exekutoren.
- Ereignisse für abgeschlossene SQL-Anfragen und zugehörige Aufgaben-, Phasen- und Aufgabenereignisse.

So starten Sie einen Cluster mit aktiviertem Rollen und Komprimieren

1. Erstellen Sie eine Konfigurationsdatei `spark-configuration.json` mit der folgenden Konfiguration.

```
[
  {
    "Classification": "spark-defaults",
    "Properties": {
      "spark.eventLog.rolling.enabled": true,
      "spark.history.fs.eventLog.rolling.maxFilesToRetain": 1
    }
  }
]
```

2. Erstellen Sie den Cluster mit der Spark-Rolling-Compaction-Konfiguration wie folgt.

```
aws emr create-cluster \
--release-label emr-6.6.0 \
--instance-type m4.large \
--instance-count 2 \
--use-default-roles \
--configurations file://spark-configuration.json
```

Berechtigungen für die Anzeige persistenter Anwendungsbensutzeroberflächen

Das folgende Beispiel zeigt die Rollenberechtigungen, die für den Zugriff auf persistente Anwendungsbensutzeroberflächen erforderlich sind. Bei Clustern mit aktivierter Runtime-Rolle können Benutzer dadurch nur auf Anwendungen zugreifen, die von derselben Benutzeridentität und Runtime-Rolle eingereicht wurden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:CreatePersistentAppUI",
        "elasticmapreduce:DescribePersistentAppUI"
      ],
      "Resource": [
        "arn:aws:elasticmapreduce:region:accountId:cluster/clusterId"
      ]
    },
    {
      "Effect": "Allow",
```

```

    "Action": [
      "elasticmapreduce:GetPersistentAppUIPresignedURL"
    ],
    "Resource": [
      "arn:aws:elasticmapreduce:region:accountId:cluster/clusterId",
      "arn:aws:elasticmapreduce:region:accountId:persistent-app-ui/*"
    ],
    "Condition": {
      "StringEqualsIfExists": {
        "elasticmapreduce:ExecutionRoleArn": [
          "executionRoleArn"
        ]
      }
    }
  }
}

```

Das folgende Beispiel zeigt die Rollenberechtigungen, die erforderlich sind, um die Einschränkungen für die Anzeige von Anwendungen in den Benutzeroberflächen für persistente Anwendungen für Cluster mit aktivierten Runtime-Rollen aufzuheben.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:CreatePersistentAppUI",
        "elasticmapreduce:DescribePersistentAppUI",
        "elasticmapreduce:AccessAllEventLogs"
      ],
      "Resource": [
        "arn:aws:elasticmapreduce:region:accountId:cluster/clusterId"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:GetPersistentAppUIPresignedURL"
      ],
      "Resource": [
        "arn:aws:elasticmapreduce:region:accountId:cluster/clusterId",

```

```

        "arn:aws:elasticmapreduce:region:accountId:persistent-app-ui/*"
    ],
    "Condition": {
        "StringEqualsIfExists": {
            "elasticmapreduce:ExecutionRoleArn": [
                "executionRoleArn"
            ]
        }
    }
}
]
}

```

Überlegungen und Einschränkungen

Der Ein-Klick-Zugriff auf persistente Anwendungsbenutzeroberflächen hat derzeit folgende Einschränkungen.

- Es wird mindestens zwei Minuten dauern, bis die Anwendungsdetails auf der Benutzeroberfläche des Spark History Servers angezeigt werden.
- Diese Funktion funktioniert nur, wenn sich das Ereignisprotokollverzeichnis für die Anwendung in HDFS befindet. Amazon EMR speichert Ereignisprotokolle standardmäßig in einem Verzeichnis von HDFS. Wenn Sie das Standardverzeichnis in ein anderes Dateisystem ändern, beispielsweise Amazon S3, funktioniert das Feature nicht.
- Diese Funktion ist derzeit nicht für EMR-Cluster mit mehreren Master-Knoten oder für EMR-Cluster mit AWS Lake Formation-Integration verfügbar.
- Um den Zugriff auf persistente Anwendungsbenutzeroberflächen mit einem Klick zu ermöglichen `CreatePersistentAppUI`, benötigen Sie Berechtigungen für Amazon EMR `DescribePersistentAppUI` und `GetPersistentAppUIPresignedURL` Aktionen. Wenn Sie einem IAM-Prinzipal die Genehmigung für diese Aktionen verweigern, dauert es ungefähr fünf Minuten, bis die Änderung der Berechtigungen wirksam wird.
- Wenn es sich bei einem Cluster um einen Cluster mit aktivierter Runtime-Rolle handelt, kann der Benutzer beim Zugriff auf den Spark History Server über die Benutzeroberfläche der persistenten App nur dann auf einen Spark-Job zugreifen, wenn der Spark-Job von einer Runtime-Rolle übermittelt wird.
- Wenn es sich bei einem Cluster um einen Cluster mit aktivierter Runtime-Rolle handelt, kann jeder Benutzer nur auf eine Anwendung zugreifen, die von derselben Benutzeridentität und Runtime-Rolle eingereicht wurde.

- Die `AccessAllEventLogs` Aktion für Amazon EMR ist erforderlich, um alle Anwendungen in persistenten Anwendungsbenutzeroberflächen für Runtime-Rollen-aktivierte Cluster anzuzeigen.
- Wenn Sie Anwendungen in einem laufenden Cluster neu konfigurieren, ist der Anwendungsverlauf nicht über das Anwendungs-UI verfügbar.
- Für jede AWS-Konto Anwendung liegt das Standardlimit für aktive Anwendungen UIs bei 200.
- Im Folgenden AWS-Regionen können Sie mit Amazon EMR 6.14.0 und höher UIs von der Konsole aus auf die Anwendung zugreifen:
 - Asien-Pazifik (Jakarta) (ap-southeast-3)
 - Europa (Spanien) (eu-south-2)
 - Asien-Pazifik (Melbourne) (ap-southeast-4)
 - Israel (Tel Aviv) (il-central-1)
 - Naher Osten (VAE) (me-central-1)
- Im Folgenden AWS-Regionen können Sie mit Amazon EMR 5.25.0 und höher UIs von der Konsole aus auf die Anwendung zugreifen:
 - USA Ost (Nord-Virginia): (us-east-1)
 - USA West (Oregon): (us-west-2)
 - Asien-Pazifik (Mumbai): (ap-south-1)
 - Asien-Pazifik (Seoul): (ap-northeast-2)
 - Asien-Pazifik (Singapur): (ap-southeast-1)
 - Asien-Pazifik (Sydney): (ap-southeast-2)
 - Asien-Pazifik (Tokyo) (ap-northeast-1)
 - Kanada (Zentral): (ca-central-1)
 - Südamerika (São Paulo) (sa-east-1)
 - Europa (Frankfurt) (eu-central-1)
 - Europa (Irland) (eu-west-1)
 - Europa (London) (eu-west-2)
 - Europa (Paris) (eu-west-3)
 - Europa (Stockholm) (eu-north-1)
 - China (Peking) (cn-north-1)
 - China (Ningxia) (cn-northwest-1)

Sehen Sie sich einen Anwendungsverlauf auf hoher Ebene in Amazon EMR an

Note

Wir empfehlen Ihnen, die persistente Anwendungsoberfläche zu verwenden, um eine bessere Benutzererfahrung zu erzielen. Dabei wird der Anwendungsverlauf bis zu 30 Tage lang gespeichert. Der auf dieser Seite beschriebene allgemeine Anwendungsverlauf ist in der neuen Amazon EMR-Konsole (<https://console.aws.amazon.com/emr>) nicht verfügbar. Weitere Informationen finden Sie unter [Persistente Anwendungsbenuzteroberflächen in Amazon EMR anzeigen](#).

Mit den Amazon-EMR-Versionen 5.8.0 bis 5.36.0 und 6.x-Versionen bis 6.8.0 können Sie auf der Registerkarte Anwendungsbenuzteroberflächen in der alten Amazon-EMR-Konsole einen allgemeinen Anwendungsverlauf anzeigen. Eine Amazon-EMR-Anwendungsbenuzteroberfläche speichert die Zusammenfassung des Anwendungsverlaufs für 7 Tage nach Abschluss eines Antrags.

Überlegungen und Einschränkungen

Berücksichtigen Sie die folgenden Einschränkungen, wenn Sie die Registerkarte Benutzteroberflächen von Anwendungen in der alten Amazon-EMR-Konsole verwenden.

- Sie können nur auf das allgemeine Anwendungsverlaufsfeature zugreifen, wenn Sie die Amazon-EMR-Versionen 5.8.0 bis 5.36.0 und 6.x-Versionen bis 6.8.0 verwenden. Mit Wirkung zum 23. Januar 2023 wird Amazon EMR den Anwendungsverlauf auf hoher Ebene für alle Versionen einstellen. Wenn Sie Amazon-EMR-Version 5.25.0 oder höher verwenden, empfehlen wir, stattdessen die persistente Anwendungsbenuzteroberfläche zu verwenden.
- Das Feature zum Anwendungsverlauf auf hoher Ebene unterstützt keine Spark-Streaming-Anwendungen.
- Der Ein-Klick-Zugriff auf persistente Anwendungsbenuzteroberflächen ist derzeit nicht für Amazon-EMR-Cluster mit mehreren Hauptknoten oder für AWS Lake Formation integrierte Amazon-EMR-Cluster verfügbar.

Beispiel: Einen übergeordneten Anwendungsverlauf anzeigen

In der folgenden Abfolge wird ein Drilldown durch eine Spark- oder YARN-Anwendung in die Aufgabedetails unter Verwendung von Anwendungsbenuzteroberflächen auf einer Clusterdetailseite in der alten Konsole veranschaulicht.

Sie können in der Liste Cluster die Option Name für einen Cluster auswählen, um Details zu diesem anzuzeigen. Um Informationen zu YARN-Container-Protokollen anzeigen zu können, müssen Sie die Protokollierung für Ihren Cluster aktivieren. Weitere Informationen finden Sie unter [Konfigurieren der Cluster-Protokollierung und des Debuggings](#). Für den Spark-Anwendungsverlauf sind die in der Zusammenfassungstabelle bereitgestellten Informationen nur eine Teilmenge der Informationen, die über die Benutzeroberfläche des Spark-History-Servers verfügbar sind.

Auf der Registerkarte Anwendungsbetzeroberflächen unter Anwendungsverlauf auf hoher Ebene können Sie eine Zeile erweitern, um die Diagnoseübersicht für eine Spark-Anwendung anzuzeigen, oder einen Anwendungs-ID-Link auswählen, um Details zu einer anderen Anwendung anzuzeigen.

Cluster: Development Cluster

Waiting Cluster ready to run steps.

Summary

Application user interfaces

Monitoring

Hardware

Configurations

Events

Steps

Bootstrap actions

Persistent application user interfaces

Applications installed on the Amazon EMR cluster publish user interfaces (UI) as web sites to monitor cluster activity. Persistent UI logs are available for 30 days after an application ends. Persistent UI don't required SSH tunneling. They are hosted off of the cluster.

Application user interface

YARN timeline server

Tez UI

Spark history server

On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require a connection via SSH tunneling. Set up SSH tunneling before accessing these application UI. [Learn more](#)

Application	User interface URL	Status
Spark History Server	http://compute-1.amazonaws.com:18080/	SSH tunnel not enabled

High-level application history

Amazon EMR collects information from YARN applications on your cluster and keeps a summary of historical information for seven days after applications have completed. [Learn more](#)

YARN applications (5)

Filter: All applications

Filter applications ...

5 applications (all loaded)

Application ID	Type	Action	Status	Start time (UTC-7)	Duration	Finish time (UTC-7)	User
▶ application_1590503538546_0005	TEZ	HIVE-62d52467-d2ac-4430-98b9-9859317f5673	Succeeded	2020-05-26 07:56 (UTC-7)	5.2 min	2020-05-26 08:02 (UTC-7)	hadoop
▶ application_1590503538546_0004	TEZ	HIVE-ea51ce39-4c0f-44f9-9613-bc8037f07710	Succeeded	2020-05-26 07:56 (UTC-7)	5.2 min	2020-05-26 08:02 (UTC-7)	hadoop
▼ application_1590503538546_0003	Spark	Spark shell	Succeeded	2020-05-26 07:50 (UTC-7)	5.5 min	2020-05-26 07:56 (UTC-7)	hadoop
Diagnostics: Succeeded							
▶ application_1590503538546_0002	Spark	Spark shell	Succeeded	2020-05-26 07:47 (UTC-7)	2.1 min	2020-05-26 07:49 (UTC-7)	hadoop
▶ application_1590503538546_0001	TEZ	HIVE-a5e557a7-dfbc-4577-87ed-4326eb7cc0f3	Succeeded	2020-05-26 07:33 (UTC-7)	5.2 min	2020-05-26 07:38 (UTC-7)	hive

Wenn Sie einen Anwendungs-ID-Link auswählen, ändert sich die Benutzeroberfläche und zeigt die YARN-Anwendungsdetails für diese Anwendung an. Auf der Registerkarte Aufträge der YARN-Anwendungsdetails können Sie den Link Beschreibung für einen Auftrag auswählen, um Details für diesen Auftrag anzuzeigen.

Cluster: Development Cluster Waiting Cluster ready to run steps.

- Summary
- Application user interfaces
- Monitoring
- Hardware
- Configurations
- Events
- Steps
- Bootstrap actions

Persistent application user interfaces

Applications installed on the Amazon EMR cluster publish user interfaces (UI) as web sites to monitor cluster activity. Persistent UI logs are available for 30 days after an application ends. Persistent UI don't required SSH tunneling. They are hosted off of the cluster.

Application user interface
YARN timeline server
Tez UI
Spark history server

On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require a connection via SSH tunneling. Set up SSH tunneling before accessing these application UI. [Learn more](#)

Application	User interface URL	Status
Spark History Server	http://...compute-1.amazonaws.com:18080/	SSH tunnel not enabled

High-level application history

[YARN applications](#) > application_1590503538546_0003 (Spark)

- Jobs
- Stages
- Executors

User: hadoop
Total uptime: 5.6 min
Completed jobs: 10

Event timeline

Jobs (10)							
Filter:		Filter jobs ...		10 jobs (all loaded)			
Job ID	Status	Description	Submitted (UTC-7)	Duration	Stages succeeded / total	Tasks succeeded / total	
9	Succeeded	collect at HoodieCopyOnWriteTable.java:329	2020-05-26 07:52 (UTC-7)	82 ms	2 / 2	4 / 4	
8	Succeeded	collect at HoodieCopyOnWriteTable.java:304	2020-05-26 07:52 (UTC-7)	1 s	1 / 1	2 / 2	
7	Succeeded	collect at AbstractHoodieWriteClient.java:140	2020-05-26 07:52 (UTC-7)	63 ms	1 / 6	1 / 4,503	
6	Succeeded	count at HoodieSparkSqlWriter.scala:257	2020-05-26 07:52 (UTC-7)	6 s	2 / 6	1,501 / 4,503	
5	Succeeded	countByKey at WorkloadProfile.java:67	2020-05-26 07:52 (UTC-7)	9 s	5 / 6	6,001 / 6,002	
4	Succeeded	countByKey at HoodieBloomIndex.java:174	2020-05-26 07:52 (UTC-7)	4 s	2 / 3	3,000 / 3,001	
3	Succeeded	collect at HoodieBloomIndex.java:218	2020-05-26 07:52 (UTC-7)	3 s	1 / 1	1 / 1	
2	Succeeded	collect at HoodieBloomIndex.java:205	2020-05-26 07:52 (UTC-7)	3 s	1 / 1	1 / 1	
1	Succeeded	countByKey at HoodieBloomIndex.java:141	2020-05-26 07:52 (UTC-7)	7 s	3 / 3	3,001 / 3,001	
0	Succeeded	isEmpty at HoodieSparkSqlWriter.scala:142	2020-05-26 07:52 (UTC-7)	8 s	1 / 1	1 / 1	

Auf der Seite der Auftragsdetails können Sie Informationen zu einzelnen Auftragsphasen erweitern und dann auf den Link Beschreibung klicken, um die Phasendetails anzuzeigen.

Cluster: Development Cluster Waiting Cluster ready to run steps.

- Summary
- Application user interfaces
- Monitoring
- Hardware
- Configurations
- Events
- Steps
- Bootstrap actions

Persistent application user interfaces

Applications installed on the Amazon EMR cluster publish user interfaces (UI) as web sites to monitor cluster activity. Persistent UI logs are available for 30 days after an application ends. Persistent UI don't required SSH tunneling. They are hosted off of the cluster.

Application user interface
YARN timeline server
Tez UI
Spark history server

On-cluster application user interfaces

On-cluster UI are available only while clusters are running. Because they are hosted on the master node, on-cluster UI require a connection via SSH tunneling. Set up SSH tunneling before accessing these application UI. [Learn more](#)

Application	User interface URL	Status
Spark History Server	http://compute-1.amazonaws.com:18080/	SSH tunnel not enabled

High-level application history

[YARN applications](#) > application_1590503538546_0003 (Spark)

- Jobs
- Stages
- Executors

Jobs > Job 9

Status: Succeeded
Completed stages: 2

► Event timeline

Stages (2)

Filter: Filter stages ...		2 stages (all loaded)							
Stage ID	Status	Description	Submitted (UTC-7)	Duration	Tasks succeeded / total	Input	Output	Shuffle read	Shuffle write
29	Completed	collect at HoodieCopyOnWriteTable.java:329	2020-05-26 07:52 (UTC-7)	20 ms	2 / 2				
Details: org.apache.spark.api.java.AbstractJavaRDDLike.collect(JavaRDDLike.scala:45) org.apache.hudi.table.HoodieCopyOnWriteTable.clean(HoodieCopyOnWriteTable.java:329) org.apache.hudi.client.HoodieCleanClient.runClean(HoodieCleanClient.java:163) org.apache.hudi.client.HoodieCleanClient.clean(HoodieCleanClient.java:98) org.apache.hudi.client.HoodieWriteClient.clean(HoodieWriteClient.java:835) org.apache.hudi.client.HoodieWriteClient.postCommit(HoodieWriteClient.java:512) org.apache.hudi.client.AbstractHoodieWriteClient.commit(AbstractHoodieWriteClient.java:157) org.apache.hudi.client.AbstractHoodieWriteClient.commit(AbstractHoodieWriteClient.java:101) org.apache.hudi.client.AbstractHoodieWriteClient.commit(AbstractHoodieWriteClient.java:92) org.apache.hudi.HoodieSparkSqlWriter\$.checkWriteStatus(HoodieSparkSqlWriter.scala:263) org.apache.hudi.HoodieSparkSqlWriter\$.write(HoodieSparkSqlWriter.scala:184) org.apache.hudi.DefaultSource.createRelation(DefaultSource.scala:91) org.apache.spark.sql.execution.datasources.SaveIntoDataSourceCommand.run(SaveIntoDataSourceCommand.scala:46) org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:70) org.apache.spark.sql.execution.command.ExecutedCommandExec.sideEffectResult(commands.scala:68) org.apache.spark.sql.execution.command.ExecutedCommandExec.doExecute(commands.scala:86) org.apache.spark.sql.execution.SparkPlan.\$anonfun\$execute\$1(SparkPlan.scala:131) org.apache.spark.sql.execution.SparkPlan.\$anonfun\$executeQuery\$1(SparkPlan.scala:156) org.apache.spark.rdd.RDDOperationScope\$.withScope(RDDOperationScope.scala:151) org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.scala:152)									
▶ 28	Completed	mapPartitionsToPair at HoodieCopyOnWriteTable.java:329	2020-05-26 07:52 (UTC-7)	31 ms	2 / 2				

Auf der Seite mit den Phasendetails können Sie die wichtigsten Kennzahlen für die Aufgaben und Ausführenden der Phase einsehen. Sie können Aufgaben- und Ausführungsprotokolle auch über die Links Protokolle anzeigen ansehen.

High-level application history

YARN applications > application_1590503538546_0003 (Spark) 

Jobs Stages Executors

Jobs > Job 9 > Stage 29 (attempt 0)

Total time across all tasks: 8 ms

Locality level summary: Process local: 2

► Event timeline

Summary metrics for 2 completed tasks

Metric 	Min	25th percentile	Median	75th percentile	Max
Duration	4 ms	4 ms	4 ms	4 ms	4 ms
GC time					
Result serialization time					
Task deserialization time	5 ms	5 ms	13 ms	13 ms	13 ms

Aggregated metrics by executor (2)

Filter: 2 executors (all loaded) 

Executor ID 	Address 	Task time	Total tasks	Failed tasks	Succeeded tasks	Blacklisted
12	ip-192-168-1-233.ec2.internal:36779 View logs	12 ms	1	0	1	No
18	ip-192-168-1-9.ec2.internal:37667 View logs	20 ms	1	0	1	No

Tasks (2)

Filter: 2 tasks (all loaded) 

ID 	Attempt	Status	Locality level	Executor ID / Host 	Launch time (UTC-7)	Duration	Task deserialization time	GC time	Result serialization time	Errors
13511	0	Succeeded	Process local	12 / ip-192-168-1-233.ec2.internal View logs	2020-05-26 07:52 (UTC-7)	12 ms	5 ms			
13512	0	Succeeded	Process local	18 / ip-192-168-1-9.ec2.internal View logs	2020-05-26 07:52 (UTC-7)	20 ms	13 ms			

Amazon EMR-Protokolldateien anzeigen

Sowohl Amazon EMR als auch Hadoop erstellen Protokolldateien, die Aufschluss über den jeweiligen Status des Clusters geben. Standardmäßig werden diese Dateien im Primärknoten im `/mnt/var/log/`-Verzeichnis gespeichert. Abhängig von der Konfiguration Ihres Clusters beim Start können diese Protokolle auch in Amazon S3 archiviert und über das grafische Debugging-Tool angezeigt werden.

Es gibt viele Arten von Protokollen, die auf dem Primärknoten gespeichert werden. Amazon EMR schreibt Schritt- und Bootstrap-Aktions- und Instance-Status-Protokolle. Apache Hadoop erstellt Protokolle mit Daten zur Verarbeitung von Aufträgen, Aufgaben und versuchten Aufgaben. Hadoop protokolliert außerdem Protokolle seiner Daemons. [Weitere Informationen zu den von Hadoop geschriebenen Protokollen finden Sie unter http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html](http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html).

Protokolldateien auf dem Primärknoten anzeigen

Die folgende Tabelle listet einige der Protokolldateien auf, die auf dem Primärknoten zu finden sind.

Ort	Beschreibung
/emr/instance-controller/log/bootstrap-Aktionen	Protokolle, die bei der Verarbeitung von Bootstrap-Aktionen geschrieben werden.
/-state-pusher mnt/var/log/hadoop	Protokolle, die vom Hadoop-Status-Push-Prozess geschrieben werden.
/emr/instance-controller/log	Instance-Controller-Protokolle.
/emr/instance-state	instance-Statusprotokolle. Diese enthalten Informationen über die CPU, den Arbeitsspeicher und Garbage Collector-Threads des Knotens.
/emr/service-nanny	Protokolle, die vom Service-Nanny-Prozess geschrieben werden.
/mnt/var/log/ <i>application</i>	Protokolle, die sich auf eine bestimmte Anwendung beziehen, wie z. B. Hadoop, Spark oder Hive.
/mnt/var/log/hadoop/steps/ <i>N</i>	Schrittprotokolle, die Informationen über die Verarbeitung des Schritts enthalten. Der Wert von <i>N</i> gibt die von Amazon EMR zugewiesene stepId an. Beispiel: Ein Cluster verfügt über zwei Schritte: s-1234ABCDEFGH und s-5678IJKLMNOP . Der erste Schritt befindet sich in /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/ und der zweite in /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/ . Die von Amazon EMR geschriebenen Schrittprotokolle lauten wie folgt.

Ort	Beschreibung
	• controller – Informationen zur Verarbeitung des Schritts. Wenn Ihr Schritt beim Laden fehlschlägt, finden Sie den Stack-Trace in diesem Protokoll. • syslog – Beschreibt die Ausführung von Hadoop-Jobs in diesem Schritt. • stderr – Der Standardfehlerkanal von Hadoop bei der Verarbeitung des Schritts. • stdout – Der Standardausgabekanal von Hadoop während der Verarbeitung des Schritts.

So zeigen Sie Protokolldateien auf dem Primärknoten mit dem AWS CLI an.

1. Verwenden Sie SSH für die Verbindung mit dem Primärknoten wie in [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#) beschrieben.
2. Navigieren Sie zu dem Verzeichnis mit den Protokolldateiinformatoren, die Sie anzeigen möchten. Die oben stehende Tabelle gibt eine Liste der verfügbaren Protokolldateien mit dem entsprechenden Speicherort an. Das folgende Beispiel zeigt den Befehl für die Navigation zum Schrittprotokoll mit einer ID, s-1234ABCDEFGH.

```
cd /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/
```

3. Verwenden Sie einen Datei-Viewer Ihrer Wahl, um die Protokolldatei anzuzeigen. Im folgenden Beispiel wird der Linux-Befehl `less` verwendet, um die Protokolldatei `controller` anzuzeigen.

```
less controller
```

In Amazon S3 archivierte Protokolldateien anzeigen

Standardmäßig archivieren mit der Konsole gestartete Amazon-EMR-Cluster Protokolldateien in Amazon S3 automatisch. Sie können einen eigenen Protokollpfad angeben, und zulassen, dass die Konsole automatisch einen Protokollpfad generiert. Für Cluster, die mit der CLI oder API gestartet wurden, müssen Sie die Archivierung des Amazon-S3-Protokolls manuell konfigurieren.

Wenn Amazon EMR für die Archivierung von Protokolldateien in Amazon S3 konfiguriert ist, speichert es die Dateien an dem von Ihnen angegebenen S3-Speicherort, im Ordner/*cluster-id*/, wo sich die Cluster-ID *cluster-id* befindet.

Die folgende Tabelle listet einige der Protokolldateien auf, die in Amazon S3 zu finden sind.

Ort	Beschreibung
//node/ <i>cluster-id</i>	Knotenprotokolle, einschließlich Bootstrap-Aktion, Instance-Status und Anwendungsprotokollen für den Knoten. Die Protokolle für jeden Knoten werden in einem Ordner gespeichert, der mit der ID der EC2 Instanz dieses Knotens beschriftet ist.
/ <i>cluster-id instance-id</i> / node// <i>application</i>	Die Protokolle, die von einzelnen Anwendungen oder Daemons, die einer Anwendung zugeordnet sind, erstellt wurden. Das Hive-Server-Protokoll befindet sich beispielsweise im Verzeichnis <i>cluster-id</i> /node/ <i>instance-id</i> /hive/hive-server.log .
//Schritte// <i>cluster-id step-id</i>	Schrittprotokolle, die Informationen über die Verarbeitung des Schritts enthalten. Der Wert von <i>step-id</i> gibt die von Amazon EMR zugewiesene Schritt-ID an. Beispiel: Ein Cluster verfügt über zwei Schritte: s-1234ABCDEFGH und s-5678IJKLMNOP . Der erste Schritt befindet sich in /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/ und der zweite in /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/ . Die von Amazon EMR geschriebenen Schrittprotokolle lauten wie folgt. • controller – Informationen zur Verarbeitung des Schritts. Wenn Ihr Schritt beim Laden

Ort	Beschreibung
	fehlschlägt, finden Sie den Stack-Trace in diesem Protokoll. • syslog – Beschreibt die Ausführung von Hadoop-Jobs in diesem Schritt. • stderr – Der Standardfehlerkanal von Hadoop bei der Verarbeitung des Schritts. • stdout – Der Standardausgabekanal von Hadoop während der Verarbeitung des Schritts.
<code>//container <i>cluster-id</i></code>	Anwendungscontainerprotokolle. Die Protokolle für jede YARN-Anwendung werden an diesen Speicherorten abgelegt.
<code>//hadoop-mapreduce/ <i>cluster-id</i></code>	Die Protokolle, die Informationen über Konfigurationsdetails und den Jobverlauf von Jobs enthalten. MapReduce

So zeigen Sie Protokolldateien an, die mit der Amazon-S3-Konsole in Amazon S3 archiviert wurden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Öffnen Sie den S3-Bucket, den Sie angegeben haben, als Sie den Cluster für die Archivierung von Protokolldateien in Amazon S3 konfiguriert haben.
3. Navigieren Sie zu der Protokolldatei, die die Informationen enthält, die angezeigt werden sollen. Die oben stehenden Tabelle gibt eine Liste der verfügbaren Protokolldateien mit dem entsprechenden Speicherort an.
4. Laden Sie das Protokolldateiobjekt herunter, um es anzuzeigen. Anweisungen finden Sie unter [Objekt herunterladen](#).

Cluster-Instances in Amazon anzeigen EC2

Um Ihnen bei der Verwaltung Ihrer Ressourcen zu helfen, EC2 ermöglicht Ihnen Amazon, Ressourcen Metadaten in Form von Tags zuzuweisen. Jedes EC2 Amazon-Tag besteht aus

einem Schlüssel und einem Wert. Mithilfe von Tags können Sie Ihre EC2 Amazon-Ressourcen auf unterschiedliche Weise kategorisieren: zum Beispiel nach Zweck, Eigentümer oder Umgebung.

Sie können die Ressourcen auf Grundlage der Tags suchen und filtern. Die Tags, die Sie Ressourcen über Ihr AWS Konto zuweisen, stehen nur Ihnen zur Verfügung. Andere Konten, die dieselbe Ressource nutzen, können Ihre Tags nicht sehen.

Amazon EMR kennzeichnet automatisch jede EC2 Instance, die es startet, mit Schlüssel-Wert-Paaren. Die Schlüssel identifizieren den Cluster und die Instance-Gruppe, zu der die Instance gehört. Dies macht es einfach, Ihre EC2 Instances zu filtern, um beispielsweise nur die Instances anzuzeigen, die zu einem bestimmten Cluster gehören, oder um alle aktuell laufenden Instances in der Instance-Gruppe für die Aufgabe anzuzeigen. Dies ist besonders nützlich, wenn Sie mehrere Cluster gleichzeitig ausführen oder eine große Anzahl von EC2 Instanzen verwalten.

Dies sind die vordefinierten Schlüssel-Wert-Paare, die Amazon EMR zuordnet:

Schlüssel	Value (Wert)	Wert-Definition
aws:elasticmapreduce:job-flow-id	<i>job-flow-identifizier</i>	Die ID des Clusters, für den die Instance bereitgestellt wird. Es wird im folgenden Format j-XXXXXXXXXXXXXX angezeigt und kann bis zu 256 Zeichen lang sein.
aws:elasticmapreduce:instance-group-role	<i>group-role</i>	Der Typ der Instance-Gruppe, eingegeben als einer der folgenden Werte: master, core oder task.

Sie können das Anzeigen und Filtern anhand der Tags ausführen, die Amazon EMR hinzufügt. Weitere Informationen finden Sie unter [Verwenden von Tags](#) im EC2 Amazon-Benutzerhandbuch. Da es sich bei den Tags von Amazon EMR um System-Tags handelt, können sie weder bearbeitet oder gelöscht werden. Aus diesem Grund sind die Abschnitte zum Anzeigen und Filtern von Tags am wichtigsten.

Note

Amazon EMR fügt der EC2 Instance Tags hinzu, wenn ihr Status auf Wird ausgeführt aktualisiert wird. Wenn zwischen dem Zeitpunkt, zu dem die EC2 Instance bereitgestellt wird, und dem Zeitpunkt, zu dem ihr Status auf Running gesetzt wird, Latenz auftritt, werden die

von Amazon EMR festgelegten Tags angezeigt, sobald die Instance gestartet wird. Wenn keine Tags angezeigt werden, warten Sie einige Minuten und aktualisieren Sie die Ansicht.

CloudWatch Ereignisse und Metriken von Amazon EMR

Verwenden Sie Ereignisse und Metriken, um die Aktivität und den Zustand eines Amazon-EMR-Clusters zu verfolgen. Ereignisse sind nützlich zur Überwachung bestimmter Vorgänge in einem Cluster, beispielsweise wenn sich der Zustand eines Clusters vom Starten zum Ausführen ändert. Metriken sind nützlich, um einen bestimmten Wert zu überwachen – beispielsweise den Prozentsatz des verfügbaren Speicherplatzes, den HDFS innerhalb eines Clusters verwendet.

Weitere Informationen zu CloudWatch Veranstaltungen finden Sie im [Amazon CloudWatch Events-Benutzerhandbuch](#). Weitere Informationen zu CloudWatch Metriken finden Sie unter [Verwenden von CloudWatch Amazon-Metriken](#) und [Erstellen von CloudWatch Amazon-Alarmen](#) im CloudWatch Amazon-Benutzerhandbuch.

Themen

- [Überwachung von Amazon EMR-Metriken mit CloudWatch](#)
- [Überwachung von Amazon EMR-Ereignissen mit CloudWatch](#)
- [Reagieren auf CloudWatch Ereignisse von Amazon EMR](#)

Überwachung von Amazon EMR-Metriken mit CloudWatch

Die Metriken werden alle fünf Minuten aktualisiert und automatisch erfasst und CloudWatch für jeden Amazon EMR-Cluster weitergeleitet. Dieses Intervall kann nicht konfiguriert werden. Für die unter angegebenen Amazon EMR-Metriken fallen keine Gebühren an CloudWatch. Diese fünfminütigen Datenpunktmetriken werden 63 Tage lang archiviert. Danach werden die Daten verworfen.

Wie verwende ich die Amazon-EMR-Metriken?

Die folgende Tabelle zeigt die häufigsten Verwendungen von Metriken, die von Amazon EMR gemeldet werden. Es handelt sich dabei um Vorschläge für den Einstieg und nicht um eine umfassende Liste. Eine Liste der gesamten Metriken, die von Amazon EMR gemeldet werden, finden Sie unter [Von Amazon EMR gemeldete Metriken in CloudWatch](#).

Wie gehe ich vor?	Relevante Metriken
Verfolgen des Cluster-Fortschritts	Sehen Sie sich die Metriken <code>RunningMapTasks</code> , <code>RemainingMapTasks</code> , <code>RunningReduceTasks</code> und <code>RemainingReduceTasks</code> an.
Erkennen von Clustern im Leerlauf	Die <code>IsIdle</code> -Metrik verfolgt, ob ein Cluster verfügbar ist, aber aktuell keine Aufgaben ausführt. Sie können einen Alarm einrichten, wenn sich der Cluster für einen bestimmten Zeitraum im Leerlauf befunden hat z. B. 30 Minuten.
Erkennen, wenn ein Knoten zu wenig Speicherplatz hat	Die <code>MRUnhealthyNodes</code> -Metrik verfolgt, wann einem oder mehreren Core- oder Aufgabenknoten der lokale Festplattenspeicher ausgeht und sie in einen <code>UNHEALTHY</code> -YARN-Status übergehen. Zum Beispiel haben Core- oder Aufgabenknoten nur noch wenig Speicherplatz zur Verfügung und sie können keine Aufgaben ausführen.
Erkennen, wenn ein Cluster zu wenig Speicherplatz hat	Die <code>HDFSUtilization</code> -Metrik überwacht die kombinierte HDFS-Kapazität des Clusters und kann eine Größenänderung des Clusters erfordern, um weitere Core-Knoten hinzuzufügen. Beispielsweise ist die HDFS-Auslastung hoch, was sich auf Aufträge und den Zustand des Clusters auswirken kann.
Erkennt, wenn ein Cluster mit reduzierter Kapazität läuft	Die <code>MRLostNodes</code> -Metrik verfolgt, wann ein oder mehrere Core- oder Aufgabenknoten nicht mit dem Hauptknoten kommunizieren können. Beispielsweise ist der Core- oder Aufgabenknoten für den Hauptknoten nicht erreichbar.

Weitere Informationen erhalten Sie unter [Der Amazon EMR-Cluster endet mit NO_SLAVE_LEFT und den Kernknoten FAILED_BY_MASTER](#) und [AWSsupport-AnalyzeEMRLogs](#).

CloudWatch Zugriffsmetriken für Amazon EMR

Sie können die Metriken, an die Amazon EMR berichtet, CloudWatch mithilfe der Amazon EMR-Konsole oder der CloudWatch Konsole anzeigen. Sie können Metriken auch mit dem CloudWatch CLI-Befehl [mon-get-stats](#) oder der CloudWatch [GetMetricStatistics](#) API abrufen. Weitere Informationen zum Anzeigen oder Abrufen von Metriken für Amazon EMR finden CloudWatch Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Console

So zeigen Sie Metriken mit der Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, für den Sie Metriken anzeigen möchten. Dadurch wird die Cluster-Detailseite geöffnet.
3. Wählen Sie auf der Cluster-Detailseite die Registerkarte Überwachung aus. Wählen Sie eine der Optionen Clusterstatus, Knotenstatus oder Ein- und Ausgaben aus, um die Berichte über den Fortschritt und den Zustand des Clusters zu laden.
4. Nachdem Sie eine Metrik zur Anzeige ausgewählt haben, können Sie jedes Diagramm vergrößern. Um den Zeitrahmen Ihres Diagramms zu filtern, wählen Sie eine vorausgefüllte Option oder wählen Sie Benutzerdefiniert.

Von Amazon EMR gemeldete Metriken in CloudWatch

In den folgenden Tabellen sind die Metriken aufgeführt, die Amazon EMR in der Konsole meldet und an die sie weiterleitet. CloudWatch

Amazon-EMR-Metriken

Amazon EMR sendet Daten für verschiedene Metriken an CloudWatch. Alle Amazon-EMR-Cluster senden automatisch Metriken in Intervallen von fünf Minuten. Die Metriken werden für zwei Wochen archiviert. Nach Ablauf dieses Zeitraums werden die Daten verworfen.

Der AWS/ElasticMapReduce-Namespace enthält die folgenden Metriken.

 Note

Amazon EMR ruft Metriken aus einem Cluster ab. Wenn die Verbindung zu einem Cluster verloren geht, werden keine Metriken gemeldet, bis der Cluster wieder verfügbar ist.

Die folgenden Metriken sind für Cluster mit Hadoop 2.x -Versionen verfügbar.

Metrik	Beschreibung
Cluster-Status	
IsIdle	Gibt an, dass ein Cluster keine Arbeiten mehr ausführt, aber unverändert aktiv ist und Kosten verursacht. Der Wert beträgt 1, wenn weder Tasks noch Aufträge ausgeführt werden, andernfalls beträgt der Wert 0. Dieser Wert wird in 5-Minuten-Intervallen geprüft. Wenn der Wert 1 beträgt, bedeutet dies, dass der Cluster zum Zeitpunkt der Prüfung ungenutzt war, aber nicht die gesamten fünf Minuten. Um Falschmeldungen zu vermeiden, sollten Sie einen Alarm auslösen, wenn dieser Wert in mehreren aufeinander folgenden 5-Minuten-Prüfungen 1 beträgt. Sie können zum Beispiel einen Alarm auslösen, wenn dieser Wert 30 Minuten oder länger 1 beträgt. Anwendungsfall: Cluster-Leistung überwachen Einheiten: boolescher Wert
ContainerAllocated	Die Anzahl der Ressourcencontainer, die von der ResourceManager zugewiesen wurden. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
ContainerReserved	Anzahl der reservierten Container. Anwendungsfall: Cluster-Fortschritt überwachen

Metrik	Beschreibung
	Einheiten: Anzahl
ContainerPending	Anzahl der Container in der Warteschlange, die noch nicht zugeordnet worden sind. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
ContainerPendingRatio	Das Verhältnis von ausstehenden Containern zu zugewiesenen Containern ($\text{ContainerPendingRatio} = \text{ContainerPending} / \text{ContainerAllocated}$). Wenn $\text{ContainerAllocated} = 0$, dann $\text{ContainerPendingRatio} = \text{ContainerPending}$. Der Wert von $\text{ContainerPendingRatio}$ steht für eine Zahl, nicht für einen Prozentsatz. Dieser Wert ist zum Skalieren von Cluster-Ressourcen anhand des Zuordnungsverhaltens des Containers hilfreich. Einheiten: Anzahl
AppsCompleted	Anzahl der an YARN übermittelten abgeschlossenen Anwendungen. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
AppsFailed	Anzahl der an YARN übermittelten Anwendungen, deren Abschluss fehlgeschlagen ist. Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl

Metrik	Beschreibung
AppsKilled	Anzahl der an YARN übermittelten Anwendungen, die beendet worden sind. Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl
AppsPending	Anzahl der an YARN übermittelten Anwendungen, die sich im ausstehenden Zustand befinden. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
AppsRunning	Anzahl der an YARN übermittelten Anwendungen, die ausgeführt werden. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
AppsSubmitted	Anzahl der an YARN übermittelten Anwendungen. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
Knotenstatus	
CoreNodesRunning	Anzahl der arbeitenden Core-Knoten. Die Datenpunkte dieser Metrik werden nur dann angegeben, wenn die zugehörige Instance-Gruppe existiert. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl

Metrik	Beschreibung
CoreNodesPending	Anzahl der Core-Knoten, die auf eine Zuordnung warten. Es müssen nicht alle angeforderten Core-Knoten sofort verfügbar sein. Diese Metrik gibt die ausstehenden Anforderungen an. Die Datenpunkte dieser Metrik werden nur dann angegeben, wenn die zugehörige Instance-Gruppe existiert. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
LiveDataNodes	Prozentsatz der Datenknoten, die Arbeit von Hadoop empfangen. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Prozent
MRTotalKnoten	Die Anzahl der Knoten, die derzeit für MapReduce Jobs verfügbar sind. Entspricht der YARN-Metrik <code>mapred.resourcemanager.TotalNodes</code>. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl Hinweis: MRTotal Nodes zählt nur die aktuell aktiven Knoten im System. YARN entfernt terminierte Knoten automatisch aus dieser Anzahl und beendet deren Verfolgung, sodass sie in der MRTotal Nodes-Metrik nicht berücksichtigt werden.

Metrik	Beschreibung
MRActiveKnoten	Die Anzahl der Knoten, auf denen derzeit MapReduce Aufgaben oder Jobs ausgeführt werden. Entspricht der YARN-Metrik <code>mapred.resourcemanager.NoOfActiveNodes</code> . Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
MRLostKnoten	Die Anzahl der Knoten, MapReduce denen der Status LOST zugewiesen wurde. Entspricht der YARN-Metrik <code>mapred.resourcemanager.NoOfLostNodes</code> . Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl
MRUnhealthyKnoten	Die Anzahl der Knoten, die für MapReduce Jobs verfügbar sind, die als FEHLERHAFT markiert sind. Entspricht der YARN-Metrik <code>mapred.resourcemanager.NoOfUnhealthyNodes</code> . Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
MRDecommissionedKnoten	Die Anzahl der Knoten, die MapReduce Anwendungen zugewiesen sind, die als STILLGELEGT markiert wurden. Entspricht der YARN-Metrik <code>mapred.resourcemanager.NoOfDecommissionedNodes</code> . Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl

Metrik	Beschreibung
MRRebootedKnoten	Die Anzahl der verfügbaren Knoten MapReduce , die neu gestartet und mit dem Status NEU GESTARTET markiert wurden. Entspricht der YARN-Metrik <code>mapred.resourcemanager.NoOfRebootedNodes</code> . Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl
MultiMasterInstanceGroupNodesRunning	Die Anzahl der zurzeit ausgeführten Master-Knoten. Anwendungsfall: Überwachen von Ausfall und Ersetzung eines Master-Knotens Einheiten: Anzahl
MultiMasterInstanceGroupNodesRunningPercentage	Der Prozentsatz der zurzeit im Verhältnis zur angeforderten Instance-Zahl für Master-Knoten ausgeführten Master-Knoten. Anwendungsfall: Überwachen von Ausfall und Ersetzung eines Master-Knotens Einheiten: Prozent
MultiMasterInstanceGroupNodesRequested	Die Anzahl der angeforderten Master-Knoten. Anwendungsfall: Überwachen von Ausfall und Ersetzung eines Master-Knotens Einheiten: Anzahl
IO	

Metrik	Beschreibung
S3 BytesWritten	Anzahl der auf Amazon S3 geschriebenen Bytes. Diese Metrik aggregiert nur MapReduce Jobs und gilt nicht für andere Workloads in Amazon EMR. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl
S3 BytesRead	Anzahl der von Amazon S3 gelesenen Bytes. Diese Metrik aggregiert nur MapReduce Jobs und gilt nicht für andere Workloads in Amazon EMR. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl
HDFSUtilization	Prozentsatz des gegenwärtig benutzten HDFS-Speichers. Anwendungsfall: Cluster-Leistung analysieren Einheiten: Prozent
HDFSBytesLesen	Anzahl der von HDFS gelesenen Byte. Diese Metrik aggregiert nur MapReduce Jobs und gilt nicht für andere Workloads in Amazon EMR. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl

Metrik	Beschreibung
HDFSBytesGeschrieben	Anzahl der auf HDFS geschriebenen Byte. Diese Metrik aggregiert nur MapReduce Jobs und gilt nicht für andere Workloads in Amazon EMR. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl
MissingBlocks	Anzahl der Blöcke, in denen HDFS keine Replicas hat. Hierbei kann es sich um beschädigte Blöcke handeln. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
CorruptBlocks	Anzahl der Blöcke, die von HDFS als beschädigt angegeben werden. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
TotalLoad	Gesamtanzahl der gleichzeitigen Datenübertragungen. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
MemoryTotalMB	Gesamtgröße des Speichers im Cluster. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
MemoryReservedMB	Größe des reservierten Speichers. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl

Metrik	Beschreibung
MemoryAvailableMB	Verfügbarer zuzuordnender Speicher. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
YARNMemoryAvailablePercentage	Der Prozentsatz des verbleibenden Speichers, der für YARN verfügbar ist ($\text{YARNMemoryAvailablePercentage} = \text{MemoryAvailable MB} / \text{MemoryTotalMB}$). Dieser Wert ist zum Skalieren von Cluster-Ressourcen anhand der YARN-Speichernutzung hilfreich. Einheiten: Prozent
MemoryAllocatedMB	Menge des dem Cluster zugeordneten Speichers. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
PendingDeletionBlocks	Anzahl der zum Löschen gekennzeichneten Blöcke. Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl
UnderReplicatedBlocks	Anzahl der Blöcke, die nochmals repliziert werden müssen. Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl

Metrik	Beschreibung
DfsPendingReplicationBlocks	Status der Blockreplikation: replizierte Blöcke, Alter der Replikationsanforderung und nicht erfolgreiche Replikationsanforderungen. Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl
CapacityRemainingGB	Gesamtbetrag der verbleibenden HDFS-Festplattenkapazität. Anwendungsfall: Cluster-Fortschritt überwachen, Cluster-Zustand überwachen Einheiten: Anzahl

Nachfolgend sind die Hadoop 1-Metriken aufgeführt:

Metrik	Beschreibung
Cluster-Status	
IsIdle	Gibt an, dass ein Cluster keine Arbeiten mehr ausführt, aber unverändert aktiv ist und Kosten verursacht. Der Wert beträgt 1, wenn weder Tasks noch Aufträge ausgeführt werden, andernfalls beträgt der Wert 0. Dieser Wert wird in 5-Minuten-Intervallen geprüft. Wenn der Wert 1 beträgt, bedeutet dies, dass der Cluster zum Zeitpunkt der Prüfung ungenutzt war, aber nicht die gesamten fünf Minuten. Um Falschmeldungen zu vermeiden, sollten Sie einen Alarm auslösen, wenn dieser Wert in mehreren aufeinander folgenden 5-Minuten-Prüfungen 1 beträgt. Sie können zum Beispiel einen Alarm auslösen, wenn dieser Wert 30 Minuten oder länger 1 beträgt.

Metrik	Beschreibung
	Anwendungsfall: Cluster-Leistung überwachen Einheiten: boolescher Wert
JobsRunning	Anzahl der Aufträge im Cluster, die gegenwärtig ausgeführt werden. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
JobsFailed	Anzahl der fehlgeschlagenen Aufträge im Cluster. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
Map/Reduce	
MapTasksRunning	Anzahl der Map-Tasks für jeden Auftrag. Wenn Sie einen Scheduler installiert haben und mehrere Aufträge ausführen, werden mehrere Grafiken erstellt. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
MapTasksRemaining	Anzahl der verbleibenden Map-Tasks für jeden Auftrag. Wenn Sie einen Scheduler installiert haben und mehrere Aufträge ausführen, werden mehrere Grafiken erstellt. Eine verbleibende Map-Task ist eine Task, die sich in keinem der folgenden Zustände befindet: Running, Killed oder Completed. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl

Metrik	Beschreibung
MapSlotsOpen	Ungenutzte Kapazität für Map-Tasks. Dies wird als die maximale Anzahl von Map-Tasks für einen bestimmten Cluster abzüglich der Gesamtanzahl der gegenwärtig ausgeführten Map-Tasks in diesem Cluster berechnet. Anwendungsfall: Cluster-Leistung analysieren Einheiten: Anzahl
RemainingMapTasksPerSlot	Das Verhältnis der insgesamt verbleibenden Map-Tasks, bezogen auf die insgesamt verfügbaren Map-Slots im Cluster. Anwendungsfall: Cluster-Leistung analysieren Einheiten: Verhältnis
ReduceTasksRunning	Anzahl der laufenden Reduce-Tasks für jeden Auftrag. Wenn Sie einen Scheduler installiert haben und mehrere Aufträge ausführen, werden mehrere Grafiken erstellt. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
ReduceTasksRemaining	Anzahl der verbleibenden Reduce-Tasks für jeden Auftrag. Wenn Sie einen Scheduler installiert haben und mehrere Aufträge ausführen, werden mehrere Grafiken erstellt. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl

Metrik	Beschreibung
ReduceSlotsOpen	Ungenutzte Kapazität für Reduce-Tasks. Dies wird als die maximale Anzahl von Reduce-Tasks für einen bestimmten Cluster abzüglich der Gesamtanzahl der gegenwärtig ausgeführten Reduce-Tasks in diesem Cluster berechnet. Anwendungsfall: Cluster-Leistung analysieren Einheiten: Anzahl
Knotenstatus	
CoreNodesRunning	Anzahl der arbeitenden Core-Knoten. Die Datenpunkte dieser Metrik werden nur dann angegeben, wenn die zugehörige Instance-Gruppe existiert. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
CoreNodesPending	Anzahl der Core-Knoten, die auf eine Zuordnung warten. Es müssen nicht alle angeforderten Core-Knoten sofort verfügbar sein. Diese Metrik gibt die ausstehenden Anforderungen an. Die Datenpunkte dieser Metrik werden nur dann angegeben, wenn die zugehörige Instance-Gruppe existiert. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
LiveDataNodes	Prozentsatz der Datenknoten, die Arbeit von Hadoop empfangen. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Prozent

Metrik	Beschreibung
TaskNodesRunning	Anzahl der arbeitenden Aufgabenknoten. Die Datenpunkte dieser Metrik werden nur dann angegeben, wenn die zugehörige Instance-Gruppe existiert. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
TaskNodesPending	Anzahl der Aufgabenknoten, die auf eine Zuordnung warten. Es müssen nicht alle angeforderten Aufgabenknoten sofort verfügbar sein. Diese Metrik gibt die ausstehenden Anforderungen an. Die Datenpunkte dieser Metrik werden nur dann angegeben, wenn die zugehörige Instance-Gruppe existiert. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl
LiveTaskTrackers	Prozentsatz der funktionierenden Task-Tracker. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Prozent
IO	
S 3 BytesWritten	Anzahl der auf Amazon S3 geschriebenen Bytes. Diese Metrik aggregiert nur MapReduce Jobs und gilt nicht für andere Workloads in Amazon EMR. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl

Metrik	Beschreibung
S3 BytesRead	Anzahl der von Amazon S3 gelesenen Bytes. Diese Metrik aggregiert nur MapReduce Jobs und gilt nicht für andere Workloads in Amazon EMR. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl
HDFSUtilization	Prozentsatz des gegenwärtig benutzten HDFS-Speichers. Anwendungsfall: Cluster-Leistung analysieren Einheiten: Prozent
HDFSBytesLesen	Anzahl der von HDFS gelesenen Byte. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl
HDFSBytesGeschrieben	Anzahl der auf HDFS geschriebenen Byte. Anwendungsfall: Cluster-Leistung analysieren, Cluster-Fortschritt überwachen Einheiten: Anzahl
MissingBlocks	Anzahl der Blöcke, in denen HDFS keine Replicas hat. Hierbei kann es sich um beschädigte Blöcke handeln. Anwendungsfall: Cluster-Zustand überwachen Einheiten: Anzahl

Metrik	Beschreibung
TotalLoad	Die aktuelle Gesamtzahl der Leser und Autoren, die von allen DataNodes in einem Cluster gemeldet wurden. Anwendungsfall: Stellen Sie fest, inwieweit ein hoher Wert zu einer schlechten Leistung bei der Auftragsausführung beitragen I/O könnte. Worker-Knoten, auf denen der DataNode Daemon ausgeführt wird, müssen auch Mapping- und Reduce-Aufgaben ausführen. Dauerhaft hohe TotalLoad Werte im Laufe der Zeit können darauf hindeuten, dass hohe Werte I/O möglicherweise zu einer schlechten Leistung beitragen. Gelegentliche Spitzen in diesem Wert sind typisch und weisen in der Regel nicht auf ein Problem hin. Einheiten: Anzahl

Cluster-Kapazitätsmetriken

Die folgenden Metriken geben die aktuelle oder Zielkapazitäten eines Clusters an. Diese Metriken sind nur verfügbar, wenn verwaltete Skalierung oder automatische Beendigung aktiviert ist.

Bei Clustern, die aus Instance-Flotten bestehen, werden die Cluster-Kapazitätsmetriken in Units gemessen. Bei Clustern, die aus Instance-Gruppen bestehen, werden die Clusterkapazitätsmetriken in Nodes oder VCPU basierend auf dem Einheitentyp gemessen, der in der Richtlinie für verwaltete Skalierung verwendet wird. Weitere Informationen finden Sie unter [Verwenden der automatischen Skalierung](#) im Verwaltungshandbuch für Amazon EMR.

Metrik	Beschreibung
TotalUnitsRequested	Die Zielgesamtanzahl von units/nodes/vCPUs in einem Cluster, wie sie durch verwaltete Skalierung bestimmt wird.
TotalNodesRequested	Einheiten: Anzahl
TotalVCPURequested	

Metrik	Beschreibung
• TotalUnitsRunning • TotalNodesRunning • TotalVCPURunning	Die aktuelle Gesamtzahl der in einem laufenden Cluster units/nodes/vCPUs verfügbaren. Wenn eine Clustergrößenänderung angefordert wird, wird diese Metrik aktualisiert, nachdem die neuen Instances hinzugefügt oder aus dem Cluster entfernt wurden. Einheiten: Anzahl
• CoreUnitsRequested • CoreNodesRequested • CoreVCPURequested	Die Zielanzahl von CORE units/nodes/vCPUs in einem Cluster, wie sie durch verwaltete Skalierung bestimmt wird. Einheiten: Anzahl
• CoreUnitsRunning • CoreNodesRunning • CoreVCPURunning	Die aktuelle Anzahl von CORE, die in einem Cluster units/nodes/vCPUs ausgeführt werden. Einheiten: Anzahl
• TaskUnitsRequested • TaskNodesRequested • TaskVCPURequested	Die Zielanzahl von TASKs units/nodes/vCPUs in einem Cluster, wie sie durch verwaltete Skalierung bestimmt wird. Einheiten: Anzahl

Metrik	Beschreibung
- TaskUnitsRunning - TaskNodesRunning - TaskVCPURunning	Die aktuelle Anzahl von TASK, die in einem Cluster units/nodes/vCPUs ausgeführt werden. Einheiten: Anzahl

Amazon EMR gibt die folgenden Metriken mit einer Granularität von einer Minute aus, wenn Sie die automatische Kündigung mithilfe einer Richtlinie zur automatischen Kündigung aktivieren. Einige Metriken sind nur für Amazon-EMR-Versionen 6.4.0 und höher verfügbar. Weitere Informationen zur automatischen Beendigung finden Sie unter [Verwenden einer automatischen Terminierungsrichtlinie für die Amazon EMR-Clusterbereinigung](#).

Metrik	Beschreibung
TotalNotebookKernels	Die Gesamtzahl der laufenden und inaktiven Notebook-Kernel auf dem Cluster. Diese Metrik ist nur für Amazon-EMR-Versionen 6.4.0 und höher verfügbar.
AutoTerminationIsClusterIdle	Gibt an, ob der Cluster verwendet wird. Der Wert 0 gibt an, dass der Cluster von einer der folgenden Komponenten aktiv verwendet wird: - Eine YARN-Anwendung - HDFS - Ein Notebook

Metrik	Beschreibung
	Eine Cluster-Benutzeroberfläche, z. B. der Spark History Server Ein Wert von 1 gibt an, dass sich der Cluster im Leerlauf befindet. Amazon EMR prüft, ob der Cluster kontinuierlich inaktiv ist (<code>AutoTerminationIsClusterIdle = 1</code>). Wenn die Leerlaufzeit eines Clusters dem <code>IdleTimeout</code>-Wert in Ihrer Richtlinie zur automatischen Kündigung entspricht, beendet Amazon EMR den Cluster.

Dimensionen für Amazon-EMR-Metriken

Die Amazon-EMR-Daten können mithilfe der folgenden Dimensionen in der folgenden Tabelle gefiltert werden.

Dimension	Beschreibung
JobFlowId	Entspricht der Cluster-ID, der eindeutigen Kennung eines Clusters mit dem Format <code>j-XXXXXXXXXXXX</code>. Sie können diesen Wert durch Klicken auf den Cluster in der Amazon-EMR-Konsole anzeigen.

Überwachung von Amazon EMR-Ereignissen mit CloudWatch

Amazon EMR verfolgt Ereignisse und speichert die Informationen für bis zu sieben Tage. Amazon EMR zeichnet Ereignisse auf, wenn sich der Status von Clustern, Instance-Gruppen, Instance-Flotten, automatischen Skalierungsrichtlinien oder Schritten ändert. Ereignisse erfassen Datum und Uhrzeit des Ereignisses, Details zu den betroffenen Elementen und andere wichtige Datenpunkte.

In der folgenden Tabelle sind Amazon-EMR-Ereignisse aufgeführt, zusammen mit dem Status oder der Statusänderung, die das Ereignis anzeigt, dem Schweregrad des Ereignisses, dem Ereignistyp,

dem Ereigniscode und den Ereignismeldungen. Amazon EMR stellt Ereignisse als JSON-Objekte dar und sendet sie automatisch an einen Event-Stream. Das JSON-Objekt ist wichtig, wenn Sie Regeln für die Ereignisverarbeitung mithilfe von CloudWatch Ereignissen einrichten, da Regeln versuchen, Mustern im JSON-Objekt zu entsprechen. Weitere Informationen finden Sie unter [Ereignisse und Ereignismuster](#) und [Amazon EMR-Ereignisse](#) im Amazon CloudWatch Events-Benutzerhandbuch.

Note

EMR gibt regelmäßig Ereignisse mit dem Ereigniscode EC2 Provisioning — Insufficient Instance Capacity aus. Diese Ereignisse treten auf, wenn Ihr Amazon EMR-Cluster während der Clustererstellung oder Größenänderung auf einen Kapazitätsfehler von Amazon EMR für Ihre Instance-Flotte oder Instance-Gruppe stößt. Ein Ereignis umfasst möglicherweise nicht alle Instance-Typen, die AZs Sie angegeben haben, da EMR nur die Instance-Typen umfasst und versucht AZs hat, Kapazität bereitzustellen, seit das letzte Ereignis mit unzureichender Kapazität ausgelöst wurde. Informationen darüber, wie Sie auf diese Ereignisse reagieren können, finden Sie unter [Reaktion auf Ereignisse mit unzureichender Instance-Kapazität im Amazon EMR-Cluster](#).

Cluster-Startereignisse

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instance-Flotten	EC2 Bereitstellung — Unzureichende Instance-Kapazität	Wir können Ihren Amazon EMR-Cluster ClusterId (ClusterName) für die Instance-Flotte nicht erstellen. InstanceFleetID Amazon EC2 hat nicht genügend Spot-Kapazität

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				für den Instance-Typ [Instance type1, Instancetype2] und nicht genügend On-Demand-Kapazität für den Instance-Typ [Instance type3, Instancetype4] in der Availability Zone[AvailabilityZone1, AvailabilityZone2] . Weitere Informationen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie in der Dokumentation .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instanzgruppen	EC2 Bereitstellung — Unzureichende Instanzkapazität	Wir können Ihren Amazon EMR-Cluster ClusterId (ClusterName) für die Instance-Gruppe nicht erstellen . InstanceGroupID Amazon EC2 hat nicht genügend Spot-Kapazität für den Instance-Typ [Instance type1, Instancetype2] und nicht genügend On-Demand-Kapazität für den Instance-Typ [Instance type3, Instancetype4] in der Availability Zone[AvailabilityZone1 , AvailabilityZone2] . Weitere

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				Informationen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie in der Dokumentation .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instance-Flotten	EC2 Bereitstellung — Nicht genügend freie Adressen im Subnetz	Wir können den Amazon EMR-Cluster <code>erClusterId</code> (<code>ClusterName</code>), den Sie für die Instance-Flotte angefordert haben, nicht erstellen, <code>InstanceFleetID</code> da das angegebene Subnetz [<code>Subnet1</code> , <code>Subnet2</code>] nicht genügend freie private IP-Adressen enthält, um Ihre Anfrage zu erfüllen. Verwenden Sie diesen <code>DescribeSubnets</code> Vorgang, um zu sehen, wie viele IP-Adressen in Ihrem Subnetz verfügbar (ungenutzt) sind. Informationen darüber,

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				wie Sie auf dieses Ereignis reagieren können, finden Sie unter Fehlercodes für die EC2 Amazon-API

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instanzgruppen	EC2 Bereitstellung — Nicht genügend freie Adressen im Subnetz	Wir können den Amazon EMR-Cluster <code>erClusterId</code> (<code>ClusterName</code>), den Sie für die Instance-Gruppe angefordert haben, nicht erstellen, <code>InstanceGroupID</code> da das angegebene Subnetz [<code>Subnet1</code> , <code>Subnet2</code>] nicht genügend freie private IP-Adressen enthält, um Ihre Anfrage zu erfüllen. Verwenden Sie diesen <code>DescribeSubnets</code> Vorgang, um zu sehen, wie viele IP-Adressen in Ihrem Subnetz verfügbar (ungenutzt) sind. Informati

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				onen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie unter Fehlercodes für die EC2 Amazon-API

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instance-Flotten	EC2 Bereitstellung — vCPU-Limit überschritten	Die Bereitstellung von InstanceFleetID im Amazon EMR-Cluster verzögert ClusterId (ClusterName) sich, da Sie das Limit für die Anzahl von v CPUs (virtuelle Verarbeitungseinheiten) erreicht haben, die den laufenden Instances in Ihrem account (accountId) zugewiesen sind. Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instanzgruppen	EC2 Bereitstellung — vCPU-Limit überschritten	Die Bereitstellung der Instance-Gruppe InstanceGroupID im Amazon EMR-Cluster verzögert ClusterId sich, da Sie das Limit für die Anzahl von vCPUs (virtuelle Verarbeitungseinheiten) erreicht haben, die den laufenden Instances in Ihrem Konto (accountId) zugewiesen sind. Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instance-Flotten	EC2 Bereitstellung — Das Limit für die Anzahl der Spot-Instances wurde überschritten	Die Bereitstellung der Instance-Flotte InstanceFleetID im Amazon EMR-Cluster verzögert ClusterID (ClusterName) sich, da Sie das Limit für die Anzahl der Spot-Instances erreicht haben, die Sie in Ihrem account (accountId) starten können. Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instanzgruppen	EC2 Bereitstellung — Das Limit für die Anzahl der Spot-Instances wurde überschritten	Die Bereitstellung der Instance-Gruppe InstanceGroupID im Amazon EMR-Cluster verzögert ClusterID (ClusterName) sich, da Sie das Limit für die Anzahl der Spot-Instances erreicht haben, die Sie in Ihrem account (accountId) starten können. Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instance-Flotten	EC2 Bereitste llung — Instanzli mit überschritten	Die Bereitste llung der Instance-Flotte InstanceFleetID im Amazon EMR-Cluster verzögert ClusterId (ClusterName) sich, da Sie das Limit für die Anzahl der Instances erreicht haben, die Sie gleichzeitig in Ihrem ausführen können. account (accountId) Weitere Informationen zu den Amazon EC2 Service Limits finden Sie unter Fehlercodes für die EC2 Amazon-API .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instanzgruppen	EC2 Bereitstellung — Instanzlimit überschritten	Die Bereitstellung der Instanzgruppe <code>InstanceGroupID</code> im Amazon EMR-Cluster verzögert <code>ClusterId</code> (<code>ClusterName</code>) sich, da Sie das Limit für die Anzahl der Instances erreicht haben, die Sie gleichzeitig in Ihrem ausführen können. <code>account</code> (<code>accountId</code>) Weitere Informationen zu den Amazon EC2 Service Limits finden Sie unter Fehlercodes für die EC2 Amazon-API .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
CREATING	WARN	Bereitstellung EMR EMR-Instanzgruppen	Keine	Der Amazon-EMR-Cluster ClusterId (ClusterName) wurde am Time erstellt und ist einsatzbereit. – oder – Der Amazon-EMR-Cluster ClusterId (ClusterName) hat die Ausführung aller ausstehenden Schritte unter Time abgeschlossen.  Note Ein Cluster im WAITING-Status kann trotzdem Aufträge bearbeiten.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
STARTING	INFO	Änderung des EMR-Cluster-Zustands	Keine	Der Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) wurde am <code>Time</code> angefordert und wird gerade erstellt.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
STARTING	INFO	Änderung des EMR-Cluster-Zustands	Keine	 Note Gilt nur für Cluster mit der Instance-Flottenkonfiguration und mehreren innerhalb von Amazon EC2 ausgewählten Availability Zones. Der Amazon-EMR-Cluster ClusterId (ClusterName) wird in Zone (AvailabilityZoneID) erstellt, die aus den angegebenen Availability-

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				Zone-Optionen ausgewählt wurde.
STARTING	INFO	Änderung des EMR-Cluster-Zustands	Keine	Der Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) begann mit der Ausführung von Schritten am <code>Time</code> .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
WAITING	INFO	Änderung des EMR-Cluster-Zustands	Keine	Der Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) wurde am <code>Time</code> erstellt und ist einsatzbereit. – oder – Der Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) hat die Ausführung aller ausstehenden Schritte unter <code>Time</code> abgeschlossen.  Note Ein Cluster im WAITING-Status kann trotzdem Aufträge bearbeiten.

 Note

Die Ereignisse mit dem Ereigniscode `EC2 provisioning - Insufficient Instance Capacity` werden regelmäßig ausgelöst, wenn Ihr EMR-Cluster während der Cluster-Erstellung oder Größenänderung auf einen Fehler von Amazon mit unzureichender Kapazität EC2 für Ihre Instance-Flotte oder Instance-Gruppe stößt. Weitere Informationen zum Umgang mit diesen Ereignissen finden Sie unter [Reaktion auf Ereignisse mit unzureichender Instance-Kapazität im Amazon-EMR-Cluster](#).

Cluster-Abbruchereignisse

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
TERMINATED	Der Schweregrad ist abhängig vom Grund für die Statusänderung, wie nachfolgend dargestellt: CRITICAL wenn der Cluster aufgrund einer der folgenden Statusänderungen beendet wurde: INTERNAL_ERROR , VALIDATION_ERROR ,	Änderung des EMR-Cluster-Zustands	Keine	Amazon EMR Cluster ClusterId (ClusterName) wurde am Time aufgrund von StateChangeReason: Code beendet.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
	INSTANCE_FAILURE , BOOTSTRAP_FAILURE oder STEP_FAILURE . • INFO wenn der Cluster aufgrund einer der folgenden Statusänderungen beendet wurde: USER_REQUEST oder ALL_STEPS_COMPLETE .			
TERMINATED_WITH_ERRORS	CRITICAL	Änderung des EMR-Cluster-Zustands	Keine	Amazon EMR Cluster ClusterId (ClusterName) wurde mit Fehlern am Time aufgrund von StateChangeReason: Code beendet.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
TERMINATE_D_WITH_ERRORS	CRITICAL	Änderung des EMR-Cluster-Zustands	Keine	Amazon EMR Cluster ClusterId (ClusterName) wurde mit Fehlern am Time aufgrund von StateChangeReason: Code beendet.

Ereignisse zur Änderung des Status der Instance-Flotte

Note

Die Konfiguration der Instance-Flotten ist nur in den Amazon-EMR-Versionen 4.8.0 und höher verfügbar, mit Ausnahme von 5.0.0 und 5.0.3.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
Von PROVISIONING bis WAITING	INFO		Keine	Bereitstellung für die Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) ist abgeschlossen.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				Die Bereitstellung startete um Time und dauerte Num Minuten. Die Instance-Flotte verfügt jetzt über eine On-Demand-Kapazität von Num und eine Spot-Kapazität von Num. Die anvisierte On-Demand-Kapazität betrug Num und die anvisierte Spot-Kapazität betrug Num.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
Von WAITING bis RESIZING	INFO		Keine	Die Größenänderung für Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) begann um Time. Die Instance-Flotte verändert ihre Größe von einer On-Demand-Kapazität von Num auf eine Zielkapazität von Num und von einer Spot-Kapazität von Num auf eine Zielkapazität von Num.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
Von RESIZING bis WAITING	INFO		Keine	Die Größenänderung für die Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) ist abgeschlossen. Die Größenänderung startete um Time und dauerte Num Minuten. Die Instance-Flotte verfügt jetzt über eine On-Demand-Kapazität von Num und eine Spot-Kapazität von Num. Die anvisierte On-Demand-Kapazität betrug Num und die anvisierte Spot-Kapazität betrug Num.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
Von RESIZING bis WAITING	INFO		Keine	Die Größenänderung der Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (Cluster Name) hat das Zeitlimit erreicht und wurde gestoppt. Die Größenänderung startete um Time und wurde nach Num Minuten gestoppt. Die Instance-Flotte verfügt jetzt über eine On-Demand-Kapazität von Num und eine Spot-Kapazität von Num. Die anvisierte On-Demand-Kapazität betrug Num und die anvisierte Spot-Kapazität betrug Num.

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
SUSPENDED	ERROR		Keine	Die Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) wurde am Time aus dem folgenden Grund gesperrt: ReasonDesc .
RESIZING	WARNING		Keine	Die Größenänderung für die Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) ist aus dem folgenden Grund blockiert: ReasonDesc .

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
WAITING oder Running	INFO		Keine	Die Größenänderung für die Instance-Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (Cluster Name) konnte nicht abgeschlossen werden, während Amazon EMR Spot-Kapazität in der Availability Zone AvailabilityZone hinzufügte. Wir haben Ihre Anfrage zur Bereitstellung zusätzlicher Spot-Kapazität storniert. Die empfohlenen Maßnahmen finden Sie unter Flexibilität der Availability Zone für einen Amazon EMR-

Status oder Statusänderung	Schweregrad	Ereignistyp	Ereigniscode	Fehlermeldung
				Cluster . Bitte versuchen Sie es erneut.
WAITING oder Running	INFO		Keine	Eine Größenänderung für beispielsweise eine Flotte InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) wurde von Entity auf Time initiiert.

Ereignisse bei der Neukonfiguration der Instance-Flotte

Status oder Statusänderung	Schweregrad	Fehlermeldung
Neukonfiguration der Instanzflotte angefordert	INFO	Ein Benutzer hat die Neukonfiguration der Instance-Flotte InstanceFleetID im Amazon EMR-Cluster ClusterId () ClusterName angefordert.
Beginn der Neukonfiguration der Instance-Flotte	INFO	Amazon EMR hat mit einer Neukonfiguration der Instance-Flotte InstanceFleetID im Amazon EMR-Cluster

Status oder Statusänderung	Schweregrad	Fehlermeldung
		ClusterId () ClusterName unter begonnen. Time
Die Neukonfiguration der Instance-Flotte ist abgeschlossen	INFO	Amazon EMR hat die Neukonfiguration der Instance-Flotte InstanceFleetID im Amazon EMR-Cluster ClusterId abgeschlossen (). ClusterName
Die Neukonfiguration der Instance-Flotte ist fehlgeschlagen	WARNING	Amazon EMR konnte die Instance-Flotte InstanceFleetID im Amazon EMR-Cluster ClusterId () ClusterName unter nicht neu konfigurieren. Time Die Neukonfiguration ist fehlgeschlagen, weil. Reason
Reversion der Neukonfiguration der Instanzflotte starten	INFO	Amazon EMR setzt die Instance-Flotte InstanceFleetID im Amazon EMR-Cluster ClusterId (ClusterName) auf die vorherige erfolgreiche Konfiguration zurück.
Die Reversion der Neukonfiguration der Instance-Flotte ist abgeschlossen	INFO	Amazon EMR hat die Wiederherstellung der Instance-Flotte InstanceFleetID im Amazon EMR-Cluster ClusterId (ClusterName) auf die vorherige erfolgreiche Konfiguration abgeschlossen.

Status oder Statusänderung	Schweregrad	Fehlermeldung
Die Reversion der Neukonfiguration der Instance-Flotte ist fehlgeschlagen	CRITICAL	Amazon EMR konnte die Instance-Flotte <code>InstanceFleetID</code> im Amazon EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) nicht auf die zuvor erfolgreiche Konfiguration unter zurücksetzen. <code>Time</code> Die Umkehrung der Neukonfiguration ist aufgrund von fehlgeschlagen. <code>Reason</code>
Die Reversion der Neukonfiguration der Instanzflotte wurde blockiert	INFO	Amazon EMR hat die Instance-Flotte <code>InstanceFleetID</code> im Amazon EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) unter vorübergehend blockiert, <code>Time</code> da sich die Instance-Flotte im Status befindet. <code>State</code>

Ereignisse zur Änderung der Größe der Instance-Flotte

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	ERROR	Spot-Provisioning-Timeout	Der Vorgang zur Größenänderung für die Instance-Flotte <code>InstanceFleetID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) konnte beim Erwerb von Spot-Kapazität in

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
			AZ AvailabilityZone nicht abgeschlossen werden. Wir haben jetzt Ihre Anfrage storniert und den Versuch beendet, zusätzliche Spot-Kapazität bereitzustellen, und die Instance-Flotte hat Spot-Kapazität von num bereitgestellt. Die Ziel-Spot-Kapazität war num. Weitere Informationen und Handlungsempfehlungen finden Sie auf der Dokumentationsseite hier . Bitte versuchen Sie es erneut.

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	ERROR	Timeout für die On-Demand-Bereitstellung	Der Vorgang zur Größenänderung für die Instance-Flotte <code>InstanceFleetID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) konnte beim Erwerb von On-Demand-Kapazität in AZ <code>AvailabilityZone</code> nicht abgeschlossen werden. Wir haben jetzt Ihre Anfrage storniert und den Versuch beendet, zusätzliche On-Demand-Kapazität bereitzustellen, und die Instance-Flotte hat On-Demand-Kapazität von <code>num</code> bereitgestellt. Die gewünschte On-Demand-Kapazität war <code>num</code> . Weitere Informationen und Handlungsempfehlungen finden Sie auf der Dokumentationsseite hier . Bitte versuchen Sie es erneut.

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	WARNING	EC2 Bereitstellung — Unzureichende Instance-Kapazität	Wir können den Vorgang zur Größenänderung für die Instance-Flotte <code>InstanceFleetID</code> im EMR-Cluster nicht abschließen, <code>ClusterId</code> (<code>ClusterName</code>) da Amazon nicht EC2 über ausreichende Spot-Kapazität für Instance-Typen [<code>Instancetype1</code> , <code>Instancetype2</code>] und über unzureichende On-Demand-Kapazität für Instance-Typen [<code>Instancetype3</code> , <code>Instancetype4</code>] in der Availability Zone verfügt. [<code>AvailabilityZone1</code>] Bisher hat die Instance-Flotte On-Demand-Kapazität von <code>num</code> bereitgestellt und die angestrebte On-Demand-Kapazität war <code>num</code> . Die bereitgestellte Spot-Kapazität ist <code>num</code>

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
			und die Ziel-Spot-Kapazität war num. Weitere Informationen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie in der Dokumentation .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	WARNING	Zeitlimit für Spot-Bereitstellung – Fortsetzung der Größenänderung	Wir stellen immer noch Spot-Kapazität für den Vorgang zur Größenänderung der Instance-Flotte bereit, der mit der time für Instance-Flotte-ID InstanceFleetID im Amazon-EMR-Cluster ClusterId (ClusterName) für [Instance type1, Instance type2] in AZ AvailabilityZone gestartet wurde. Für den vorherigen Vorgang zur Größenänderung, der am time gestartet wurde, ist der Timeout-Zeitraum abgelaufen, sodass Amazon EMR die Bereitstellung von Spot-Kapazität eingestellt hat, nachdem die angeforderte Anzahl von Instances zu Ihrer Instance-Flotte hinzugefügt wurden. Weitere Informationen finden Sie auf der

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
			Dokumentationsseite hier .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	WARNING	Timeout für On-Demand-Bereitstellung – Fortsetzung der Größenänderung	Wir stellen weiterhin On-Demand-Kapazität für den Vorgang zur Größenänderung der Instance-Flotte bereit, der mit der <code>time</code> für Instance-Flotte-ID <code>InstanceFleetID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) für [<code>InstanceType1</code> , <code>InstanceType2</code>] in AZ <code>AvailabilityZone</code> gestartet wurde. Für den vorherigen Vorgang zur Größenänderung, der am <code>time</code> gestartet wurde, ist der Timeout-Zeitraum abgelaufen, sodass Amazon EMR die Bereitstellung von On-Demand-Kapazität eingestellt hat, nachdem die angeforderten <code>num</code> Instances zu Ihrer Instance-Flotte hinzugefügt wurden. Weitere Informationen finden Sie auf der

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
			Dokumentationsseite hier .
Größe der EMR-Instance-Flotte ändern	WARNING	EC2 Bereitstellung — Unzureichende freie Adresse im Subnetz	Wir können die Größenänderung für die Instance-Flotte <code>InstanceFleetID</code> im Amazon EMR-Cluster nicht abschließen, <code>ClusterId</code> (<code>ClusterName</code>) da das angegebene Subnetz [<code>Subnet1</code> , <code>Subnet2</code>] nicht genügend freie private IP-Adressen enthält, um Ihre Anfrage zu bearbeiten. Verwenden Sie diesen <code>DescribeSubnets</code> Vorgang, um zu sehen, wie viele IP-Adressen in Ihrem Subnetz verfügbar (ungenutzt) sind. Informationen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie unter Fehlercodes für die EC2 Amazon-API .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	WARNING	EC2 Bereitstellung — vCPU-Limit überschritten	Die Größenänderung der Instance-Flotte <code>InstanceFleetID</code> im Amazon EMR-Cluster verzögert <code>ClusterName</code> sich, da Sie das Limit für die Anzahl der V CPUs (virtuelle Verarbeitungseinheiten) erreicht haben, die den laufenden Instances in Ihrem zugewiesen sind. <code>account</code> (<code>accountId</code>) Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	WARNING	EC2 Bereitstellung — Das Limit für die Anzahl der Spot-Instances wurde überschritten	Die Bereitstellung der Instance-Flotte <code>InstanceFleetID</code> im Amazon EMR-Cluster verzögert <code>ClusterID</code> (<code>ClusterName</code>) sich, da Sie das Limit für die Anzahl der Spot-Instances erreicht haben, die Sie in Ihrem account (<code>accountId</code>) starten können. Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instance-Flotte ändern	WARNING	EC2 Bereitstellung — Instanzlimit überschritten	Die Bereitstellung der Instance-Flotte <code>InstanceFleetID</code> im Amazon EMR-Cluster verzögert <code>ClusterID</code> (<code>ClusterName</code>) sich, da Sie das Limit für die Anzahl der On-Demand-Instances, die Sie in Ihrem account (<code>accountId</code>) ausführen können, erreicht haben. Weitere Informationen zu Fehlercodes für die EC2 Amazon-API .

Note

Die Timeout-Ereignisse für die Bereitstellung werden ausgelöst, wenn Amazon EMR die Bereitstellung von Spot- oder On-Demand-Kapazität für die Flotte nach Ablauf des Timeouts beendet. Weitere Informationen zum Umgang mit diesen Ereignissen finden Sie unter [Reaktion auf Timeout-Ereignisse zur Größenänderung der Amazon-EMR-Cluster-Instance-Flotte](#).

Instance-Gruppen-Ereignisse

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Von RESIZING bis Running	INFO	Keine	Die Größenänderung für die Instance-

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
			Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) ist abgeschlossen. Sie verfügt jetzt über eine Instance-Anzahl von Num. Die Größenänderung startete um Time und dauerte Num Minuten bis zum Abschluss.
Von RUNNING bis RESIZING	INFO	Keine	Eine Größenänderung für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) begann bei Time. Die Größenänderung erfolgt von einer Instance-Anzahl von Num auf Num.

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
SUSPENDED	ERROR	Keine	Die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) wurde am Time aus dem folgenden Grund gesperrt: ReasonDesc .
RESIZING	WARNING	Keine	Die Größenänderung für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) ist aus dem folgenden Grund blockiert: ReasonDesc .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instanzgruppe ändern	WARNING	EC2 Bereitstellung — Unzureichende Instanzkapazität	Wir können den Vorgang zur Größenänderung, der bei <code>time</code> für die Instanzgruppe <code>InstanceGroupID</code> im EMR-Cluster gestartet wurde, nicht abschließen, <code>ClusterId</code> (<code>ClusterName</code>) da Amazon nicht genügend Spot/On Demand Kapazität für den Instance-Typ <code>[Instance type]</code> in der Availability Zone <code>EC2</code> hat. <code>[AvailabilityZone1]</code> Bisher hatte die Instance-Gruppe eine Anzahl laufender Instances von <code>num</code> und die Anzahl der angeforderten Instances betrug <code>num</code> . Weitere Informationen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie in der Dokumentation .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instanzgruppe ändern	WARNING	EC2 Bereitstellung — Unzureichende freie Adresse im Subnetz	Wir können die Größenänderung für die Instance-Gruppe InstanceGroupID im Amazon EMR-Cluster nicht abschließen, ClusterId (ClusterName) da das angegebene Subnetz [Subnet1, Subnet2] nicht genügend freie private IP-Adressen enthält, um Ihre Anfrage zu bearbeiten. Verwenden Sie diesen DescribeSubnets Vorgang, um zu sehen, wie viele IP-Adressen in Ihrem Subnetz verfügbar (ungenutzt) sind. Informationen darüber, wie Sie auf dieses Ereignis reagieren können, finden Sie unter Fehlercodes für die EC2 Amazon-API .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instanzgruppe ändern	WARNING	EC2 Bereitstellung — vCPU-Limit überschritten	Die Größenänderung der Instanzgruppe InstanceGroupID im Amazon EMR-Cluster verzögert ClusterName sich, da Sie das Limit für die Anzahl von v CPUs (virtuelle Verarbeitungseinheiten) erreicht haben, die den laufenden Instances in Ihrem zugewiesen sind. account (accountId) Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instanzgruppe ändern	WARNING	EC2 Bereitstellung — Das Limit für die Anzahl der Spot-Instances wurde überschritten	Die Bereitstellung der Instance-Gruppe <code>InstanceGroupID</code> im Amazon EMR-Cluster verzögert <code>ClusterID</code> (<code>ClusterName</code>) sich, da Sie das Limit für die Anzahl der Spot-Instances erreicht haben, die Sie in Ihrem account (<code>accountId</code>) starten können. Weitere Informationen finden Sie unter Fehlercodes für die EC2 Amazon-API .

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung
Größe der EMR-Instanzgruppe ändern	WARNING	EC2 Bereitstellung — Instanzlimit überschritten	Die Bereitstellung der Instanzgruppe <code>InstanceGroupID</code> im Amazon EMR-Cluster verzögert <code>ClusterID</code> (<code>ClusterName</code>) sich, da Sie das Limit für die Anzahl der On-Demand-Instances, die Sie in Ihrem account (<code>accountId</code>) ausführen können, erreicht haben. Weitere Informationen zu Fehlercodes für die EC2 Amazon-API .
Von RUNNING bis RESIZING	INFO	Keine	Eine Größenänderung für die Instance-Gruppe <code>InstanceGroupID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>ClusterName</code>) wurde von Entity auf Time initiiert.

 Note

Ab Amazon-EMR-Version 5.21.0 können Sie Cluster-Konfigurationen überschreiben und zusätzliche Konfigurationsklassifikationen für jede Instance-Gruppe in einem ausgeführten Cluster angeben. Dazu verwenden Sie die Amazon EMR-Konsole, das AWS Command Line

Interface (AWS CLI) oder das AWS SDK. Weitere Informationen finden Sie unter [Angeben einer Konfiguration für eine Instance-Gruppe in einem aktiven Cluster](#).

In der folgenden Tabelle sind die Amazon-EMR-Ereignisse für den Rekonfigurationsvorgang aufgeführt, zusammen mit dem Zustand oder der Zustandsänderung, die das Ereignis anzeigt, dem Schweregrad des Ereignisses und den Ereignismeldungen.

Status oder Statusänderung	Schweregrad	Fehlermeldung
RUNNING	INFO	Eine Neukonfiguration für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) wurde vom Benutzer um Time initiiert. Die Version der angeforderten Konfiguration ist Num.
Von RECONFIGURING bis Running	INFO	Der Neukonfigurationsvorgang für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) ist abgeschlossen. Die Rekonfiguration startete um Time und benötigte Num Minuten bis zum Abschluss. Die aktuelle Konfigurationsversion ist Num.
Von RUNNING bis RECONFIGURING in	INFO	Eine Neukonfiguration für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) begann um Time. Sie konfiguriert

Status oder Statusänderung	Schweregrad	Fehlermeldung
		von Versionsnummer Num bis Versionsnummer Num.
RESIZING	INFO	Die Rekonfigurationsoperation auf Konfigurationsversion Num für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) wurde vorübergehend gesperrt, da sich die Instance-Gruppe Time in State befindet.
RECONFIGURING	INFO	Die Größenänderungsoperation auf Instance-Anzahl Num für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) wurde vorübergehend auf Time gesperrt, da die Instance-Gruppe sich in State befindet.
RECONFIGURING	WARNING	Die Rekonfigurationsoperation für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) schlug um Time fehl. Bis zum Fehlschlagen dauerte es Num Minuten. Die fehlgeschlagene Konfigurationsversion ist Num.

Status oder Statusänderung	Schweregrad	Fehlermeldung
RECONFIGURING	INFO	Konfigurationen werden in der zuvor funktionierenden Version Num für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) auf Time wiederhergestellt. Die neue Konfigurationsversion ist Num.
Von RECONFIGURING bis Running	INFO	Konfigurationen wurden in der zuvor funktionierenden Version Num für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) auf Time wiederhergestellt. Die neue Konfigurationsversion ist Num.
Von RECONFIGURING bis SUSPENDED	CRITICAL	Fehler beim Zurücksetzen auf die vorherige erfolgreiche Version Num für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (ClusterName) unter Time.

Auto-Scaling-Richtlinienereignisse

Status oder Statusänderung	Schweregrad	Fehlermeldung
PENDING	INFO	Der Instance-Gruppe InstanceGroupID in

Status oder Statusänderung	Schweregrad	Fehlermeldung
		Amazon-EMR-Cluster <code>ClusterId</code> (<code>Cluster Name</code>) wurde um <code>Time</code> eine Auto-Scaling-Richtlinie hinzugefügt. Die Ausrüstung der Richtlinie ist noch anhängig. – oder – Die Auto Scaling-Richtlinie für die Instance-Gruppe <code>InstanceGroupID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>Cluster Name</code>) wurde um <code>Time</code> aktualisiert. Die Ausrüstung der Richtlinie ist noch anhängig.
ATTACHED	INFO	Die Auto Scaling-Richtlinie für die Instance-Gruppe <code>InstanceGroupID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>Cluster Name</code>) wurde um <code>Time</code> angefügt.
DETACHED	INFO	Die Auto Scaling-Richtlinie für die Instance-Gruppe <code>InstanceGroupID</code> im Amazon-EMR-Cluster <code>ClusterId</code> (<code>Cluster Name</code>) wurde um <code>Time</code> getrennt.

Status oder Statusänderung	Schweregrad	Fehlermeldung
FAILED	ERROR	Die Auto-Scaling-Richtlinie für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (Cluster Name) konnte nicht angefügt werden und schlug um Time fehl. – oder – Die Auto Scaling-Richtlinie für die Instance-Gruppe InstanceGroupID im Amazon-EMR-Cluster ClusterId (Cluster Name) konnte nicht getrennt werden und schlug um Time fehl.

Schritt-Ereignisse

Status oder Statusänderung	Schweregrad	Fehlermeldung
PENDING	INFO	Der Schritt StepID (StepName) wurde dem Amazon-EMR-Cluster ClusterId (ClusterName) um Time hinzugefügt und seine Ausführung steht noch aus.
CANCEL_PENDING	WARN	Der Schritt StepID (StepName) im Amazon-EMR-Cluster ClusterId

Status oder Statusänderung	Schweregrad	Fehlermeldung
RUNNING	INFO	(ClusterName) wurde um Time storniert und die Stornierung steht noch aus. Schritt StepID (StepName) im Amazon-EMR-Cluster ClusterId (ClusterName) wurde um Time gestartet.
COMPLETED	INFO	Die Ausführung des Schritts StepID (StepName) im Amazon-EMR-Cluster ClusterId (ClusterName) wurde um Time abgeschlossen. Der Schritt begann um Time mit der Ausführung und dauerte Num Minuten bis zum Abschluss.
CANCELLED	WARN	Die Abbruchsaufforderung war für den Cluster-Schritt StepID (StepName) im Amazon-EMR-Cluster ClusterId (ClusterName) um Time erfolgreich und der Schritt wurde abgebrochen.
FAILED	ERROR	Der Schritt StepID (StepName) im Amazon-EMR-Cluster ClusterId (ClusterName) ist um Time fehlgeschlagen.

Fehlerhafte Ereignisse beim Austausch von Knoten

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung	
Austausch eines fehlerhaften Amazon EMR-Knotens	INFO	Ein fehlerhafter Core-Knoten wurde erkannt	Amazon EMR hat festgestellt, dass sich die Kerninstanz [instanceID (Instance Name)] InstanceGroup/Fleet im Amazon EMR-Cluster clusterID (ClusterName) befindet. UNHEALTHY Amazon EMR wird versuchen, die Instance wiederherzustellen oder ordnungsgemäß zu ersetzen. UNHEALTHY	
Austausch eines fehlerhaften Amazon EMR-Knotens	INFO	Der Kernknoten ist defekt — der Austausch ist deaktiviert	Amazon EMR hat festgestellt, dass sich die Kerninstanz [instanceID (Instance Name)] InstanceGroup/Fleet	

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung	
			t im Amazon EMR-Cluster (clusterID) (ClusterName) befindet. UNHEALTHY Aktivieren Sie den ordnungsgemäßen Austausch ungesunder Kernknoten in Ihrem Cluster, damit Amazon EMR die UNHEALTHY Instances ordnungsgemäß ersetzt, falls sie nicht wiederhergestellt werden können.	

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung	
Austausch eines fehlerhaften Amazon EMR-Knotens	WARN	Fehlerhafter Kernknoten wurde nicht ersetzt	Amazon EMR kann Ihre <i>UNHEALTHY</i> Core-Instance <i>[instanceID (Instance Name)] InstanceGroup/Fleet</i> im Amazon EMR-Cluster aus gutem Grund <i>clusterID (ClusterName)</i> Grund nicht ersetzen.  Note Der Grund, warum Amazon EMR Ihren Core-Node nicht ersetzen kann, hängt von Ihrem Szenario ab. Ein	

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung	
			Grund dafür, dass Amazon EMR einen Knoten nicht löschen kann, ist beispielsweise, dass ein Cluster keine verbleibenden Kernknoten haben würde.	

Ereignistyp	Schweregrad	Ereigniscode	Fehlermeldung	
Austausch eines fehlerhaften Amazon EMR-Knotens	INFO	Ein fehlerhafter Kernknoten wurde wiederhergestellt	Amazon EMR hat Ihre UNHEALTHY Core-Instances [instanceID (Instance Name)] InstanceGroup/Fleet im Amazon EMR-Cluster wiederhergestellt clusterID (ClusterName)	

Weitere Informationen zum Austausch fehlerhafter Knoten finden Sie unter [Ersetzen](#) fehlerhafter Knoten.

Ereignisse mit der Amazon-EMR-Konsole anzeigen

Für jeden Cluster können Sie eine einfache Liste der Ereignisse im Detailbereich anzeigen, der die Ereignisse in der Reihenfolge ihres Auftretens auflistet. Sie können auch alle Ereignisse für alle Cluster in einer Region in absteigender Reihenfolge ihres Auftretens anzeigen.

Wenn Sie nicht möchten, dass ein Benutzer alle Cluster-Ereignisse für eine Region sehen kann, erstellen Sie eine Anweisung, die die Berechtigung ("Effect": "Deny") für die Aktion `elasticmapreduce:ViewEventsFromAllClustersInConsole` ablehnt. Fügen Sie diese Anweisung einer Richtlinie hinzu, die dem Benutzer zugeordnet ist.

So zeigen Sie Ereignisse für alle Cluster in einer Region mit der Konsole an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Events aus.

Um Ereignisse für einen bestimmten Cluster mit der Konsole anzuzeigen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann einen Cluster aus.
3. Um alle Ihre Ereignisse anzuzeigen, wählen Sie auf der Cluster-Detailseite die Registerkarte Ereignisse aus.

Reagieren auf CloudWatch Ereignisse von Amazon EMR

In diesem Abschnitt werden verschiedene Möglichkeiten beschrieben, wie Sie auf umsetzbare Ereignisse reagieren können, die Amazon EMR als CloudWatch Ereignismeldungen ausgibt. Zu den Möglichkeiten, auf Ereignisse zu reagieren, gehören das Erstellen von Regeln, das Einstellen von Alarmen und andere Reaktionen. Die folgenden Abschnitte enthalten Links zu Verfahren und empfohlenen Antworten auf häufig auftretende Ereignisse.

Themen

- [Regeln für Amazon EMR-Ereignisse erstellen mit CloudWatch](#)
- [Alarme für CloudWatch Metriken aus Amazon EMR einrichten](#)
- [Reaktion auf Ereignisse mit unzureichender Instance-Kapazität im Amazon-EMR-Cluster](#)
- [Reaktion auf Timeout-Ereignisse zur Größenänderung der Amazon-EMR-Cluster-Instance-Flotte](#)

Regeln für Amazon EMR-Ereignisse erstellen mit CloudWatch

Amazon EMR sendet Ereignisse automatisch an einen CloudWatch Event-Stream. Sie können Regeln erstellen, die nach einem bestimmten Muster auf Ereignisse zutreffen, und Sie können die Ereignisse an Ziele weiterleiten, um entsprechende Maßnahmen zu ergreifen, z. B. E-Mail-Benachrichtigungen senden. Muster werden mit dem JSON-Objekt abgeglichen. Weitere Informationen zu Amazon EMR-Veranstaltungsdetails finden Sie unter [Amazon EMR-Ereignisse](#) im Amazon CloudWatch Events-Benutzerhandbuch.

Informationen zum Einrichten von CloudWatch Ereignisregeln finden Sie unter [Eine CloudWatch Regel erstellen, die bei einem Ereignis ausgelöst wird](#).

Alarme für CloudWatch Metriken aus Amazon EMR einrichten

Amazon EMR überträgt Metriken an Amazon CloudWatch. Als Reaktion darauf können Sie Alarme CloudWatch für Ihre Amazon EMR-Metriken einrichten. Sie können beispielsweise einen Alarm so konfigurieren, CloudWatch dass Sie jedes Mal, wenn die HDFS-Auslastung über 80% steigt, eine E-Mail erhalten. Eine ausführliche Anleitung finden Sie unter [Einen CloudWatch Alarm erstellen oder bearbeiten](#) im CloudWatch Amazon-Benutzerhandbuch.

Reaktion auf Ereignisse mit unzureichender Instance-Kapazität im Amazon-EMR-Cluster

Übersicht

Amazon-EMR-Cluster geben den Ereigniscode `EC2 provisioning - Insufficient Instance Capacity` zurück, wenn die ausgewählte Availability Zone nicht über genügend Kapazität verfügt, um Ihre Anfrage zum Clusterstart oder zur Größenänderung zu erfüllen. Das Ereignis wird regelmäßig sowohl bei Instance-Gruppen als auch bei Instance-Flotten ausgelöst, wenn Amazon EMR wiederholt auf Ausnahmen mit unzureichender Kapazität stößt und Ihre Bereitstellungsanforderung für einen Cluster-Start oder eine Cluster-Größenänderung nicht erfüllen kann.

Auf dieser Seite wird beschrieben, wie Sie am besten auf diesen Ereignistyp reagieren können, wenn er für Ihren EMR-Cluster auftritt.

Empfohlene Reaktion auf ein Ereignis mit unzureichender Kapazität

Es wird empfohlen, dass Sie auf ein Ereignis mit unzureichender Kapazität mit einer der folgenden Methoden reagieren:

- Warten Sie, bis die Kapazität wiederhergestellt ist. Die Kapazität ändert sich häufig, sodass sich eine Ausnahme mit unzureichender Kapazität von selbst erholen kann. Ihre Cluster beginnen oder beenden die Größenänderung, sobald EC2 Amazon-Kapazität verfügbar ist.
- Alternativ können Sie Ihren Cluster beenden, Ihre Instance-Typ-Konfigurationen ändern und einen neuen Cluster mit der aktualisierten Cluster-Konfigurationsanforderung erstellen. Weitere Informationen finden Sie unter [Flexibilität der Availability Zone für einen Amazon EMR-Cluster](#).

Sie können auch Regeln oder automatische Reaktionen auf ein Ereignis mit unzureichender Kapazität einrichten, wie im nächsten Abschnitt beschrieben.

Automatisierte Wiederherstellung nach einem Ereignis mit unzureichender Kapazität

Sie können eine Automatisierung als Reaktion auf Amazon-EMR-Ereignisse erstellen, z. B. solche mit Ereigniscode `EC2 provisioning - Insufficient Instance Capacity`. Die folgende AWS Lambda Funktion beendet beispielsweise einen EMR-Cluster mit einer Instanzgruppe, die On-Demand-Instances verwendet, und erstellt dann einen neuen EMR-Cluster mit einer Instanzgruppe, die andere Instance-Typen als die ursprüngliche Anfrage enthält.

Die folgenden Bedingungen lösen den automatisierten Prozess aus:

- Das Ereignis „unzureichende Kapazität“ wird seit mehr als 20 Minuten für Primär- oder Core-Knoten ausgelöst.
- Der Cluster befindet sich nicht im Status `READY` oder `WAITING`. Weitere Informationen zu EMR-Cluster-Status finden Sie unter [Verstehen des Cluster-Lebenszyklus](#).

Note

Wenn Sie einen automatisierten Prozess für eine Ausnahme mit unzureichender Kapazität erstellen, sollten Sie berücksichtigen, dass das Ereignis „unzureichende Kapazität“ wiederherstellbar ist. Die Kapazität ändert sich häufig und Ihre Cluster setzen die Größenänderung fort oder beginnen mit dem Betrieb, sobald EC2 Amazon-Kapazität verfügbar ist.

Example Funktion zur Reaktion auf ein Ereignis mit unzureichender Kapazität

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone

INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE = "EMR Instance Group Provisioning"
INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE = (
    "EC2 provisioning - Insufficient Instance Capacity"
)
ALLOWED_INSTANCE_TYPES_TO_USE = [
    "m5.xlarge",
```



```

    "c5.xlarge",
    "m5.4xlarge",
    "m5.2xlarge",
    "t3.xlarge",
]
CLUSTER_START_ACCEPTABLE_STATES = ["WAITING", "RUNNING"]
CLUSTER_START_SLA = 20

CLIENT = boto3.client("emr", region_name="us-east-1")

# checks if the incoming event is 'EMR Instance Fleet Provisioning' with eventCode 'EC2
# provisioning - Insufficient Instance Capacity'
def is_insufficient_capacity_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE
        )

# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceGroupType could be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]
    clusterCreationTime = describeClusterResponse["Cluster"]["Status"]["Timeline"][
        "CreationDateTime"
    ]
]
clusterState = describeClusterResponse["Cluster"]["Status"]["State"]

now = datetime.datetime.now()
now = now.replace(tzinfo=timezone.utc)
isClusterStartSlaBreached = clusterCreationTime < now - datetime.timedelta(
    minutes=CLUSTER_START_SLA
)

# Check if instance group receiving Insufficient capacity exception is CORE or
PRIMARY (MASTER),
# and it's been more than 20 minutes since cluster was created but the cluster
state and the cluster state is not updated to RUNNING or WAITING
if (
    (instanceGroupType == "CORE" or instanceGroupType == "MASTER")

```

```

        and isClusterStartSlaBreached
        and clusterState not in CLUSTER_START_ACCEPTABLE_STATES
    ):
        return True
    else:
        return False

# Choose item from the list except the exempt value
def choice_excluding(exempt):
    for i in ALLOWED_INSTANCE_TYPES_TO_USE:
        if i != exempt:
            return i

# Create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceGroupType could be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]

    # Following two lines assumes that the customer that created the cluster already
    # knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # Select new instance types to include in the new createCluster request
    instanceTypeForMaster = (
        instanceTypesFromOriginalRequestMaster
        if instanceGroupType != "MASTER"
        else choice_excluding(instanceTypesFromOriginalRequestMaster)
    )
    instanceTypeForCore = (
        instanceTypesFromOriginalRequestCore
        if instanceGroupType != "CORE"
        else choice_excluding(instanceTypesFromOriginalRequestCore)
    )

    print("Starting to create cluster...")
    instances = {
        "InstanceGroups": [
            {
                "InstanceRole": "MASTER",
                "InstanceCount": 1,
                "InstanceType": instanceTypeForMaster,

```

```

        "Market": "ON_DEMAND",
        "Name": "Master",
    },
    {
        "InstanceRole": "CORE",
        "InstanceCount": 1,
        "InstanceType": instanceTypeForCore,
        "Market": "ON_DEMAND",
        "Name": "Core",
    },
]
}
response = CLIENT.run_job_flow(
    Name="Test Cluster",
    Instances=instances,
    VisibleToAllUsers=True,
    JobFlowRole="EMR_EC2_DefaultRole",
    ServiceRole="EMR_DefaultRole",
    ReleaseLabel="emr-6.10.0",
)

return response["JobFlowId"]

# Terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")

def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response

def lambda_handler(event, context):
    if is_insufficient_capacity_event(event):
        print(
            "Received insufficient capacity event for instanceGroup, clusterId: "
            + event["detail"]["clusterId"]
        )

    describeClusterResponse = describe_cluster(event)

```

```

    shouldTerminateCluster = is_cluster_eligible_for_termination(
        event, describeClusterResponse
    )
    if shouldTerminateCluster:
        terminate_cluster(event)

        clusterId = create_cluster(event)
        print("Created a new cluster, clusterId: " + clusterId)
    else:
        print(
            "Cluster is not eligible for termination, clusterId: "
            + event["detail"]["clusterId"]
        )

else:
    print("Received event is not insufficient capacity event, skipping")

```

Reaktion auf Timeout-Ereignisse zur Größenänderung der Amazon-EMR-Cluster-Instance-Flotte

Übersicht

Amazon-EMR-Cluster senden [Ereignisse](#) aus, während sie die Größenänderung für z. B. Flottencluster ausführen. Die Timeout-Ereignisse für die Bereitstellung werden ausgelöst, wenn Amazon EMR die Bereitstellung von Spot- oder On-Demand-Kapazität für die Flotte nach Ablauf des Timeouts beendet. Die Dauer des Timeouts kann vom Benutzer im Rahmen der [Größenänderungsspezifikationen](#) für die Instance-Flotten konfiguriert werden. In Szenarien mit aufeinanderfolgenden Größenänderungen für dieselbe Instance-Flotte gibt Amazon EMR die Ereignisse `Spot provisioning timeout - continuing resize` oder `On-Demand provisioning timeout - continuing resize` aus, wenn das Timeout für den aktuellen Größenänderungsvorgang abläuft. Dann beginnt es mit der Bereitstellung von Kapazität für die nächste Größenänderung der Flotte.

Reagieren auf Timeout-Ereignisse zur Größenänderung der Instanceflotte

Es wird empfohlen, dass Sie auf ein Bereitstellungs-Timeout-Ereignis mit einer der folgenden Methoden reagieren:

- Greifen Sie die [Größenänderungsspezifikationen](#) wieder auf und versuchen Sie erneut, die Größe zu ändern. Da sich die Kapazität häufig ändert, wird die Größe Ihrer Cluster erfolgreich angepasst,

sobald EC2 Amazon-Kapazität verfügbar ist. Wir empfehlen unseren Kunden, niedrigere Werte für die Timeout-Dauer für Jobs zu konfigurieren, die strengere Anforderungen stellen. SLAs

- Alternativ können Sie entweder:
 - Einen neuen Cluster mit diversifizierten Instance-Typen auf der Grundlage von [bewährten Methoden wie der Flexibilität von Instances und Availability Zones](#) starten oder
 - Einen Cluster mit On-Demand-Kapazität starten
- Für das Ereignis „Timeout bei der Bereitstellung – Fortsetzung der Größenänderung“ können Sie zusätzlich warten, bis die Größenänderungsvorgänge verarbeitet sind. Amazon EMR verarbeitet weiterhin sequentiell die für die Flotte ausgelösten Größenänderungsvorgänge, wobei die konfigurierten Größenänderungsspezifikationen eingehalten werden.

Sie können auch Regeln oder automatische Reaktionen auf dieses Ereignis einrichten, wie im nächsten Abschnitt beschrieben.

Automatisierte Wiederherstellung nach einem Bereitstellungs-Timeout-Ereignis

Mit dem `Spot Provisioning timeout`-Ereigniscode können Sie als Reaktion auf Amazon-EMR-Ereignisse eine Automatisierung erstellen. Die folgende AWS Lambda -Funktion fährt beispielsweise einen EMR-Cluster mit einer Instance-Flotte herunter, die Spot Instances für Aufgabenknoten verwendet, und erstellt dann einen neuen EMR-Cluster mit einer Instance-Flotte, die vielfältigere Instance-Typen als die ursprüngliche Anfrage enthält. In diesem Beispiel löst das für Aufgabenknoten ausgegebene `Spot Provisioning timeout` Ereignis die Ausführung der Lambda-Funktion aus.

Example Beispielfunktion zur Reaktion auf ein **Spot Provisioning timeout**-Ereignis

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone

SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE = "EMR Instance Fleet Resize"
SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE = (
    "Spot Provisioning timeout"
)

CLIENT = boto3.client("emr", region_name="us-east-1")
```

```
# checks if the incoming event is 'EMR Instance Fleet Resize' with eventCode 'Spot
provisioning timeout'
def is_spot_provisioning_timeout_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE
        )

# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceFleetType could be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # Check if instance fleet receiving Spot provisioning timeout event is TASK
    if (instanceFleetType == "TASK"):
        return True
    else:
        return False

# create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceFleetType cloud be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # the following two lines assumes that the customer that created the cluster
    already knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # select new instance types to include in the new createCluster request
    instanceTypesForTask = [
        "m5.xlarge",
        "m5.2xlarge",
        "m5.4xlarge",
        "m5.8xlarge",
        "m5.12xlarge"
    ]
```

```

print("Starting to create cluster...")
instances = {
    "InstanceFleets": [
        {
            "InstanceFleetType": "MASTER",
            "TargetOnDemandCapacity": 1,
            "TargetSpotCapacity": 0,
            "InstanceTypeConfigs": [
                {
                    'InstanceType': instanceTypesFromOriginalRequestMaster,
                    "WeightedCapacity": 1,
                }
            ]
        },
        {
            "InstanceFleetType": "CORE",
            "TargetOnDemandCapacity": 1,
            "TargetSpotCapacity": 0,
            "InstanceTypeConfigs": [
                {
                    'InstanceType': instanceTypesFromOriginalRequestCore,
                    "WeightedCapacity": 1,
                }
            ]
        },
        {
            "InstanceFleetType": "TASK",
            "TargetOnDemandCapacity": 0,
            "TargetSpotCapacity": 100,
            "LaunchSpecifications": {},
            "InstanceTypeConfigs": [
                {
                    'InstanceType': instanceTypesForTask[0],
                    "WeightedCapacity": 1,
                },
                {
                    'InstanceType': instanceTypesForTask[1],
                    "WeightedCapacity": 2,
                },
                {
                    'InstanceType': instanceTypesForTask[2],
                    "WeightedCapacity": 4,
                },
            ],
        }
    ]
}

```

```

        {
            'InstanceType': instanceTypesForTask[3],
            "WeightedCapacity":8,
        },
        {
            'InstanceType': instanceTypesForTask[4],
            "WeightedCapacity":12,
        }
    ],
    "ResizeSpecifications": {
        "SpotResizeSpecification": {
            "TimeoutDurationMinutes": 30
        }
    }
}

]
}

response = CLIENT.run_job_flow(
    Name="Test Cluster",
    Instances=instances,
    VisibleToAllUsers=True,
    JobFlowRole="EMR_EC2_DefaultRole",
    ServiceRole="EMR_DefaultRole",
    ReleaseLabel="emr-6.10.0",
)

return response["JobFlowId"]

```

terminated the cluster using clusterId received in an event

```

def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")

```

```

def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response

```

```

def lambda_handler(event, context):
    if is_spot_provisioning_timeout_event(event):
        print(

```



```

        "Received spot provisioning timeout event for instanceFleet, clusterId: "
        + event["detail"]["clusterId"]
    )

    describeClusterResponse = describe_cluster(event)

    shouldTerminateCluster = is_cluster_eligible_for_termination(
        event, describeClusterResponse
    )
    if shouldTerminateCluster:
        terminate_cluster(event)

        clusterId = create_cluster(event)
        print("Created a new cluster, clusterId: " + clusterId)
    else:
        print(
            "Cluster is not eligible for termination, clusterId: "
            + event["detail"]["clusterId"]
        )

    else:
        print("Received event is not spot provisioning timeout event, skipping")

```

Cluster-Anwendungsmetriken mithilfe von Ganglia mit Amazon EMR anzeigen

Ganglia ist mit Amazon EMR-Versionen zwischen 4.2 und 6.15 verfügbar. Ganglia ist ein Open-Source-Projekt. Es handelt sich um ein skalierbares, verteiltes System zur Überwachung von Clustern und Grids, das zugleich die Auswirkungen auf die Leistung minimiert. Wenn Sie Ganglia in Ihrem Cluster aktivieren, können Sie Berichte erstellen und die Leistung des Clusters als Ganzes betrachten. Ebenso können Sie die Leistung einzelner Knoten-Instances überprüfen. Ganglia ist außerdem zur Aufnahme und Visualisierung von Hadoop- und Spark-Metriken konfiguriert. Weitere Informationen finden Sie unter [Ganglia](#) im Handbuch zu Amazon-EMR-Versionen.

Protokollieren AWS EMR EMR-API-Aufrufen mit AWS CloudTrail

AWS EMR ist in einen Dienst integriert [AWS CloudTrail](#), der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service ausgeführten Aktionen bereitstellt. CloudTrail erfasst alle API-Aufrufe für AWS EMR als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS EMR-Konsole und Codeaufrufen für die AWS EMR-API-Operationen. Anhand der von

gesammelten Informationen können Sie die Anfrage CloudTrail, die an AWS EMR gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

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 Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto, wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem AWS-Region. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter [Arbeiten mit dem CloudTrail Ereignisverlauf](#). Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder [CloudTrailLake-Event-Datenspeicher](#).

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS Management Console sind regionsübergreifend. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten AWS-Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere Informationen zu Trails finden Sie unter [Erstellen eines Trails für Ihr AWS-Konto](#) und [Erstellen eines Trails für eine Organisation](#) im AWS CloudTrail -Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preise](#). Informationen zu Amazon-S3-Preisen finden Sie unter [Amazon S3 – Preise](#).

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail [Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format](#). ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von [erweiterten Ereignisselektoren](#) auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter [Arbeiten mit AWS CloudTrail Lake](#) im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die [Preisoption](#) aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preise](#).

AWS EMR-Datenereignisse in CloudTrail

[Datenereignisse](#) liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in ein Amazon-S3-Objekt). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preisgestaltung](#).

Sie können Datenereignisse für die AWS EMR-Ressourcentypen mithilfe der CloudTrail Konsole oder CloudTrail API-Operationen protokollieren. AWS CLI Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter [Protokollieren von Datenereignissen mit dem AWS Management Console](#) und [Protokollieren von Datenereignissen mit dem AWS Command Line Interface](#) im AWS CloudTrail -Benutzerhandbuch.

In der folgenden Tabelle sind die AWS EMR-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, der aus

der Liste Datenereignistyp auf der CloudTrail Konsole ausgewählt werden kann. In der Wertspalte `resources.type` wird der `resources.type` Wert angezeigt, den Sie bei der Konfiguration erweiterter Event-Selektoren mithilfe von oder angeben würden. AWS CLI CloudTrail APIs In der CloudTrail Spalte APIs Protokolierte Daten werden die API-Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Weitere Informationen zu diesen API-Vorgängen finden Sie in der [Amazon EMR WAL \(EMRWAL\)](#) CLI-Referenz. Amazon EMR protokolliert einige Daten-API-Operationen, bei CloudTrail denen es sich um HBase Systemvorgänge handelt, die Sie nie direkt aufrufen. Diese Operationen sind nicht in der EMRWAL-CLI-Referenz enthalten.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten wurden protokolliert APIs CloudTrail
Amazon EMR-Write-Ahead-Log-Arbeitsbereich	AWS::EMRWAL::Workspace	- GetCurrentWALTime - ListTagsForResource - Liste WALs - ListWorkspaces - TrimWal - Vollständig WALFlush

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den Feldern `eventName`, `readOnly` und `resources.ARN` filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Kontingenten finden Sie unter [AdvancedFieldSelector](#) in der AWS CloudTrail -API-Referenz.

AWS EMR-Managementveranstaltungen in CloudTrail

[Managementereignisse](#) bieten Informationen zu Verwaltungsvorgängen, die an Ressourcen in Ihrem AWS-Konto ausgeführt werden. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail protokolliert standardmäßig Verwaltungsereignisse.

AWS EMR protokolliert alle Operationen der AWS EMR-Steuerebene als Verwaltungsereignisse. Eine Liste der Operationen auf der AWS EMR-Steuerungsebene, die AWS EMR protokolliert CloudTrail, finden Sie in der [AWS EMR-API-Referenz](#).

AWS Beispiele für EMR-Ereignisse

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API-Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die RunJobFlowAktion demonstriert.

```
{
  "Records": [
    {
      "eventVersion": "1.01",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/temporary-user-xx-7M",
        "accountId": "123456789012",
        "userName": "temporary-user-xx-7M"
      },
      "eventTime": "2018-03-31T17:59:21Z",
      "eventSource": "elasticmapreduce.amazonaws.com",
      "eventName": "RunJobFlow",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.1",
      "userAgent": "aws-sdk-java/unknown-version Linux/xx Java_HotSpot(TM)_64-Bit_Server_VM/xx",
      "requestParameters": {
        "tags": [
          {
            "value": "prod",
            "key": "domain"
          },
          {
            "value": "us-west-2",
            "key": "realm"
          },
          {
            "value": "VERIFICATION",
            "key": "executionType"
          }
        ]
      }
    }
  ]
}
```

```

    ],
    "instances":{
      "slaveInstanceType":"m5.xlarge",
      "ec2KeyName":"emr-integtest",
      "instanceCount":1,
      "masterInstanceType":"m5.xlarge",
      "keepJobFlowAliveWhenNoSteps":true,
      "terminationProtected":false
    },
    "visibleToAllUsers":false,
    "name":"MyCluster",
    "ReleaseLabel":"emr-5.16.0"
  },
  "responseElements":{
    "jobFlowId":"j-2WDJCGEG4E6AJ"
  },
  "requestID":"2f482daf-b8fe-11e3-89e7-75a3d0e071c5",
  "eventID":"b348a38d-f744-4097-8b2a-e68c9b424698"
},
...additional entries
]
}

```

Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter [CloudTrailDatensatzinhalte](#).

Bewährte Methoden zur EMR-Observabilität

EMR Observability umfasst einen umfassenden Überwachungs- und Managementansatz für AWS EMR-Cluster. Die Grundlage bildet Amazon CloudWatch als primärer Monitoring-Service, ergänzt durch EMR Studio und Tools von Drittanbietern wie Prometheus und Grafana für mehr Transparenz. In diesem Dokument untersuchen wir spezifische Aspekte der Cluster-Observability:

1. [Spark-Observability](#) (GitHub) — In Bezug auf die Spark-Benutzeroberfläche haben Sie in Amazon EMR drei Optionen.
2. [Spark-Fehlerbehebung](#) (GitHub) — Lösungen für Fehler.
3. [EMR-Cluster-Überwachung](#) (GitHub) — Überwachung der Cluster-Leistung.
4. [Fehlerbehebung bei EMR](#) (GitHub) — Identifizieren, diagnostizieren und lösen Sie häufig auftretende EMR-Clusterprobleme.

5. [Kostenoptimierung](#) (GitHub) — In diesem Abschnitt werden die bewährten Methoden für die Ausführung kostengünstiger Workloads beschrieben.

Tool zur Leistungsoptimierung für Apache Spark-Anwendungen

1. AWS Das [EMR Advisor-Tool](#) analysiert Spark-Ereignisprotokolle, um maßgeschneiderte Empfehlungen zur Optimierung der EMR-Clusterkonfigurationen, zur Leistungssteigerung und zur Kostensenkung zu geben. Es nutzt historische Daten und schlägt ideale Executor-Größen und Infrastruktureinstellungen vor, was eine effizientere Ressourcennutzung und eine insgesamt verbesserte Cluster-Leistung ermöglicht.
2. Das [Amazon CodeGuru Profiler-Tool](#) hilft Entwicklern dabei, Leistungsengpässe und Ineffizienzen in ihren Spark-Anwendungen zu identifizieren, indem es Laufzeitdaten sammelt und analysiert. Das Tool lässt sich nahtlos in bestehende Spark-Anwendungen integrieren, erfordert nur eine minimale Einrichtung und bietet über die AWS Konsole detaillierte Einblicke in die CPU-Auslastung, Speichermuster und Leistungs-Hotspots.

Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen

Sie können die Anzahl der für einen Amazon EMR-Cluster verfügbaren EC2 Amazon-Instances automatisch oder manuell anpassen, um auf Workloads mit unterschiedlichen Anforderungen zu reagieren. Um die automatische Skalierung zu verwenden, haben Sie zwei Optionen. Sie können Amazon EMR Managed Scaling aktivieren oder eine benutzerdefinierte Richtlinie für Auto Scaling erstellen. Die folgende Tabelle beschreibt die Unterschiede zwischen den Optionen.

	Amazon EMR Managed Scaling	Benutzerdefinierte automatische Skalierung
Skalieren von Richtlinien und Regeln	Keine Richtlinie erforderlich. Amazon EMR verwaltet die Aktivität von Auto Scaling durch kontinuierliche Auswertung von Cluster-Metriken und optimierte Skalierungsentscheidungen.	Sie müssen die Richtlinien und Regeln für das Auto Scaling definieren und verwalten, z. B. die spezifischen Bedingungen, die Skalierungsaktivitäten, Evaluierungszeiträume, Ruhephasen usw. auslösen.

	Amazon EMR Managed Scaling	Benutzerdefinierte automatische Skalierung
Unterstützte Versionen für Amazon EMR	Amazon-EMR-Version 5.30.0 und höher (außer Amazon-EMR-Version 6.0.0)	Amazon-EMR-Version 4.0.0 und höher
Unterstützte Clusterzusammenstellung	Instance-Gruppen oder Instance-Flotten	Nur Instance-Gruppen
Konfiguration von Skalierungsgrenzen	Skalierungsgrenzwerte werden für den gesamten Cluster konfiguriert.	Skalierungslimits können nur für jede Instance-Gruppe konfiguriert werden.
Häufigkeit der Auswertung von Metriken	Alle 5 bis 10 Sekunden Eine häufigere Auswertung von Metriken ermöglicht es Amazon EMR, präzisere Skalierungsentscheidungen zu treffen.	Sie können die Auswertungszeiträume nur in Fünf-Minuten-Schritten definieren.
Unterstützte Anwendungen	Es werden nur YARN-Anwendungen wie Spark, Hadoop, Hive, Flink unterstützt. Amazon EMR Managed Scaling unterstützt keine Anwendungen, die nicht auf YARN basieren, wie Presto oder. HBase	Sie können auswählen , welche Anwendungen unterstützt werden, wenn Sie die Regeln für eine automatische Skalierung definieren.

Überlegungen

- Ein Amazon-EMR-Cluster besteht immer aus einem oder drei Primärknoten. Sobald Sie den Cluster zum ersten Mal konfiguriert haben, können Sie nur Core- und Aufgabenknoten skalieren. Sie können die Anzahl der Primärknoten für den Cluster nicht skalieren.

- Bei Instance-Gruppen werden Rekonfigurations- und Größenänderungsvorgänge nacheinander und nicht gleichzeitig ausgeführt. Wenn Sie eine Neukonfiguration initiieren, während die Größe einer Instance-Gruppe geändert wird, beginnt die Neukonfiguration, sobald die Instance-Gruppe die laufende Größenänderung abgeschlossen hat. Umgekehrt, wenn Sie eine Größenänderung einleiten, während eine Instance-Gruppe ihre Neukonfiguration durchführt.

Verwenden der verwalteten Skalierung in Amazon EMR

Important

Wir empfehlen dringend, die neueste Amazon EMR-Version (Amazon EMR 7.9.0) für verwaltete Skalierung zu verwenden. In einigen frühen Versionen kann es zu zeitweiligen Anwendungsausfällen oder Verzögerungen bei der Skalierung kommen. Amazon EMR hat dieses Problem mit den 5.x-Versionen 5.30.2, 5.31.1, 5.32.1, 5.33.1 und höher sowie mit den 6.x-Versionen 6.1.1, 6.2.1, 6.3.1 und höher behoben. Weitere Informationen zur Region und Release-Verfügbarkeit finden Sie unter [Verwaltete Skalierungsverfügbarkeit](#)

Übersicht

Mit den Amazon-EMR-Versionen 5.30.0 und höher (außer Amazon EMR 6.0.0) können Sie die von Amazon EMR Managed Scaling aktivieren. Managed Scaling hilft Ihnen, die Anzahl der Instances oder Einheiten in Ihrem Cluster basierend auf der Workload automatisch zu erhöhen oder zu verringern. Amazon EMR wertet Cluster-Metriken kontinuierlich aus, um Skalierungsentscheidungen zu treffen, die Ihre Cluster für Kosten und Geschwindigkeit optimieren. Verwaltete Skalierung ist für Cluster verfügbar, die entweder aus Instance-Gruppen oder Instance-Flotten bestehen.

Verwaltete Skalierungsverfügbarkeit

- Im Folgenden AWS-Regionen ist die von Amazon EMR verwaltete Skalierung mit Amazon EMR 6.14.0 und höher verfügbar:
 - Europa (Spanien) (eu-south-2)
- Im Folgenden AWS-Regionen ist die von Amazon EMR verwaltete Skalierung mit Amazon EMR 5.30.0 und 6.1.0 und höher verfügbar:
 - USA Ost (Nord-Virginia): (us-east-1)
 - USA Ost (Ohio): (us-east-2)

- USA West (Oregon): (us-west-2)
- USA West (Nordkalifornien) (us-west-1)
- Afrika (Kapstadt) (af-south-1)
- Asien-Pazifik (Hongkong) (ap-east-1)
- Asien-Pazifik (Mumbai): (ap-south-1)
- Asien-Pazifik (Hyderabad) (ap-south-2)
- Asien-Pazifik (Seoul): (ap-northeast-2)
- Asien-Pazifik (Singapur): (ap-southeast-1)
- Asien-Pazifik (Sydney): (ap-southeast-2)
- Asien-Pazifik (Jakarta) (ap-southeast-3)
- Asien-Pazifik (Tokyo) (ap-northeast-1)
- Asien-Pazifik (Osaka) (ap-northeast-3)
- Kanada (Zentral): (ca-central-1)
- Südamerika (São Paulo) (sa-east-1)
- Europa (Frankfurt) (eu-central-1)
- Europa (Zürich) (eu-central-2)
- Europa (Irland) (eu-west-1)
- Europa (London) (eu-west-2)
- Europa (Mailand) (eu-south-1)
- Europa (Paris) (eu-west-3)
- Europa (Stockholm) (eu-north-1)
- Israel (Tel Aviv) (il-central-1)
- Naher Osten (VAE) (me-central-1)
- China (Peking) (cn-north-1)
- China (Ningxia) (cn-northwest-1)
- AWS GovCloud (US-Ost) (-1) us-gov-east
- AWS GovCloud (US-West) (us-gov-west-1)
- Amazon EMR Managed Scaling funktioniert nur mit YARN-Anwendungen wie Spark, Hadoop, Hive und Flink. Es unterstützt keine Anwendungen, die nicht auf YARN basieren, wie Presto und HBase

Verwaltete Skalierungsparameter

Sie müssen die folgenden Parameter für die verwaltete Skalierung konfigurieren. Das Limit gilt nur für die Kern- und Aufgabenknoten. Der Primärknoten kann nach der Erstkonfiguration nicht skaliert werden.

- **Minimum (MinimumCapacityUnits)** — Die untere Grenze der zulässigen EC2 Kapazität in einem Cluster. Sie wird durch vCPU-Kerne oder Instances für Instance-Gruppen gemessen. Sie wird in Einheiten für Instance-Flotten gemessen.
- **Maximum (MaximumCapacityUnits)** — Die obere Grenze der zulässigen EC2 Kapazität in einem Cluster. Sie wird durch vCPU-Kerne oder Instances für Instance-Gruppen gemessen. Sie wird in Einheiten für Instance-Flotten gemessen.
- **On-Demand-Limit (MaximumOnDemandCapacityUnits) (optional)** — Die Obergrenze der zulässigen EC2 Kapazität für den On-Demand-Markttyp in einem Cluster. Wenn dieser Parameter nicht angegeben wird, wird der Standardwert MaximumCapacityUnits verwendet.
 - Dieser Parameter wird verwendet, um die Kapazitätszuweisung zwischen On-Demand- und Spot Instances aufzuteilen. Wenn Sie beispielsweise den Minimalparameter auf 2 Instances, den Maximalparameter auf 100 Instances und das On-Demand-Limit auf 10 Instances festlegen, skaliert Amazon EMR Managed Scaling auf bis zu 10 On-Demand-Instances und weist die verbleibende Kapazität Spot Instances zu. Weitere Informationen finden Sie unter [Knotenzuweisungsszenarien](#).
- **Maximale Anzahl an Kernknoten (MaximumCoreCapacityUnits) (optional)** — Die Obergrenze der zulässigen EC2 Kapazität für den Core-Knotentyp in einem Cluster. Wenn dieser Parameter nicht angegeben wird, wird der Standardwert MaximumCapacityUnits verwendet.
 - Dieser Parameter wird verwendet, um die Kapazitätszuweisung zwischen Core- und Aufgabenknoten aufzuteilen. Wenn Sie beispielsweise den Minimalparameter auf 2 Instances, das Maximum auf 100 Instances und den maximalen Core-Knoten auf 17 Instances festlegen, skaliert Amazon EMR Managed Scaling auf bis zu 17 Core-Knoten und weist die verbleibenden 83 Instances Aufgabenknoten zu. Weitere Informationen finden Sie unter [Knotenzuweisungsszenarien](#).

Weitere Informationen zu verwalteten Skalierungsparametern finden Sie unter [ComputeLimits](#).

Hinweise zu Amazon EMR Managed Scaling

- Managed Scaling wird in limitierten Versionen AWS-Regionen und Amazon EMR-Versionen unterstützt. Weitere Informationen finden Sie unter [Verwaltete Skalierungsverfügbarkeit](#).
- Sie müssen die folgenden Parameter für die Amazon EMR Managed Scaling konfigurieren. Weitere Informationen finden Sie unter [Verwaltete Skalierungsparameter](#).
- Um verwaltete Skalierung verwenden zu können, muss der Metrics-Collector-Prozess in der Lage sein, eine Verbindung zum öffentlichen API-Endpunkt für die verwaltete Skalierung in API Gateway herzustellen. Wenn Sie einen privaten DNS-Namen mit verwenden Amazon Virtual Private Cloud, funktioniert die verwaltete Skalierung nicht ordnungsgemäß. Um sicherzustellen, dass die verwaltete Skalierung funktioniert, empfehlen wir, dass Sie eine der folgenden Aktionen ausführen:
 - Entfernen Sie den VPC-Endpunkt der API-Gateway-Schnittstelle aus Ihrer Amazon VPC.
 - Folgen Sie den Anweisungen unter [Warum erhalte ich den Fehler HTTP 403 Forbidden, wenn ich von einer VPC APIs aus eine Verbindung zu meinem API Gateway herstelle?](#) um die Einstellung für den privaten DNS-Namen zu deaktivieren.
 - Starten Sie Ihren Cluster stattdessen in einem privaten Subnetz. Weitere Informationen finden Sie im Thema [Private Subnetze](#).
- Wenn Ihre YARN-Jobs während des Herunterskalierens zeitweise langsam sind und die YARN Ressource Manager-Protokolle zeigen, dass die meisten Ihrer Knoten während dieser Zeit auf der Negativliste standen, können Sie den Schwellenwert für die Außerbetriebnahme anpassen.

Reduzieren Sie den `spark.blacklist.decommissioning.timeout` von einer Stunde auf eine Minute, um den Knoten für andere ausstehende Container verfügbar zu machen, um die Aufgabenverarbeitung fortzusetzen.

Sie sollten auch einen höheren Wert `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` festlegen, um sicherzustellen, dass Amazon EMR das Beenden des Knotens nicht erzwingt, solange die längste „Spark-Task“ noch auf dem Knoten läuft. Die aktuelle Standardeinstellung ist 60 Minuten, was bedeutet, dass YARN den Container nach 60 Minuten auf jeden Fall beendet, sobald der Knoten in den Stilllegungszustand übergeht.

Die folgende Protokollzeile von YARN Resource Manager zeigt Knoten, die dem Status „Außerbetriebnahme“ hinzugefügt wurden:

```
2021-10-20 15:55:26,994 INFO  
org.apache.hadoop.YARN.server.resourcemanager.DefaultAMSPProcessor
```

```
(IPC Server handler 37 on default port 8030): blacklist are updated in
Scheduler.blacklistAdditions: [ip-10-10-27-207.us-west-2.compute.internal,
ip-10-10-29-216.us-west-2.compute.internal, ip-10-10-31-13.us-
west-2.compute.internal, ... , ip-10-10-30-77.us-west-2.compute.internal],
blacklistRemovals: []
```

Erfahren Sie mehr darüber wie [Amazon EMR bei der Außerbetriebnahme von Knoten mit YARN Ablehnungsliste](#) integriert wird, zu [Fällen, in denen Knoten in Amazon EMR abgelehnt werden können](#), und zur [Konfiguration des Verhaltens bei der Außerbetriebnahme von Spark-Knoten](#).

- Eine übermäßige Auslastung von EBS-Volumes kann zu Problemen bei der verwalteten Skalierung führen. Wir empfehlen, das EBS-Volumen unter einer Auslastung 90 % zu halten. Weitere Informationen finden Sie unter [Speicheroptionen und Verhalten von Instances in Amazon EMR](#).
- CloudWatch Amazon-Metriken sind entscheidend für den Betrieb der von Amazon EMR verwalteten Skalierung. Wir empfehlen Ihnen, die CloudWatch Amazon-Metriken genau zu beobachten, um sicherzustellen, dass keine Daten fehlen. Weitere Informationen darüber, wie Sie CloudWatch Alarme konfigurieren können, um fehlende Messwerte zu erkennen, finden Sie unter [CloudWatch Amazon-Alarme verwenden](#).
- Verwaltete Skalierungsvorgänge auf Clustern der Versionen 5.30.0 und 5.30.1, ohne dass Presto installiert ist, können zu Anwendungsausfällen führen oder dazu führen, dass eine einheitliche Instance-Gruppe oder Instance-Flotte unverändert im Status ARRESTED bleibt, insbesondere wenn auf einen Herunterskalierungsvorgang schnell ein Skalierungsvorgang folgt.

Um dieses Problem zu umgehen, wählen Sie Presto als zu installierende Anwendung, wenn Sie einen Cluster mit den Amazon-EMR-Versionen 5.30.0 und 5.30.1 erstellen, auch wenn Ihr Auftrag Presto nicht benötigt.

- Wenn Sie den maximalen Core-Knoten und das On-Demand-Limit für Amazon EMR Managed Scaling festlegen, sollten Sie die Unterschiede zwischen Instance-Gruppen und Instance-Flotten berücksichtigen. Jede Instance-Gruppenkonfiguration besteht aus Knotentypen aus demselben Instance-Typ und derselben Kaufoption für Instances: On-Demand oder Spot. Für jede Instance-Flotte geben Sie bis zu fünf Instance-Typen an, die als On-Demand- und Spot Instances bereitgestellt werden können. Weitere Informationen finden Sie unter [Erstellen eines Clusters mit Instance-Flotten oder einheitlichen Instance-Gruppen](#), [Instance Flotten Optionen](#) und [Knotenzuweisungsszenarien](#).
- Wenn Sie bei Amazon EMR 5.30.0 und höher die standardmäßige ausgehende Regel Alle zulassen auf 0.0.0.0/ für die Master-Sicherheitsgruppe entfernen, müssen Sie eine Regel hinzufügen, die ausgehende TCP-Konnektivität zu Ihrer Sicherheitsgruppe für den Servicezugriff am Port 9443 zulässt. Ihre Sicherheitsgruppe für den Servicezugang muss auch eingehenden

TCP-Verkehr auf Port 9443 von der Master-Sicherheitsgruppe zulassen. Weitere Informationen über die Konfiguration von Sicherheitsgruppen finden Sie unter [Von Amazon EMR verwaltete Sicherheitsgruppe für die primäre Instance \(private Subnetze\)](#).

- Sie können es verwenden AWS CloudFormation , um Amazon EMR Managed Scaling zu konfigurieren. Weitere Informationen finden Sie unter [AWS::EMR::Cluster](#) im AWS CloudFormation -Benutzerhandbuch.
- Wenn Sie Spot-Knoten verwenden, sollten Sie die Verwendung von Knotenbezeichnungen in Betracht ziehen, um zu verhindern, dass Amazon EMR Anwendungsprozesse entfernt, wenn Amazon EMR Spot-Knoten entfernt. Weitere Informationen zu Knotenbezeichnungen finden Sie unter [Task-Knoten](#).
- Die Knotenkennzeichnung wird in Amazon EMR-Versionen 6.15 oder niedriger standardmäßig nicht unterstützt. Weitere Informationen finden Sie unter [Grundlegendes zu Knotentypen: Primär-, Kern- und Aufgabenknoten](#).
- Wenn Sie Amazon EMR-Versionen 6.15 oder niedriger verwenden, können Sie Knotenbezeichnungen nur nach Knotentyp zuweisen, z. B. Kern- und Aufgabenknoten. Wenn Sie jedoch Amazon EMR Version 7.0 oder höher verwenden, können Sie Node-Labels nach Knotentyp und Markttyp konfigurieren, z. B. On-Demand und Spot.
- Wenn die Nachfrage nach Anwendungsprozessen steigt und die Nachfrage nach Ausführern sinkt, wenn Sie den Anwendungsprozess auf Kernknoten beschränkt haben, können Sie bei derselben Größenänderung wieder Kernknoten hinzufügen und Aufgabenknoten entfernen. Weitere Informationen finden Sie unter [Grundlegendes zu Strategien und Szenarien für die Knotenzuweisung](#).
- Amazon EMR kennzeichnet Aufgabenknoten nicht, sodass Sie die YARN-Eigenschaften nicht so einstellen können, dass Anwendungsprozesse nur für Aufgabenknoten eingeschränkt werden. Wenn Sie jedoch Markttypen als Knotenbezeichnungen verwenden möchten, können Sie die SPOT Bezeichnungen ON_DEMAND oder für die Platzierung von Antragsprozessen verwenden. Wir empfehlen nicht, Spot-Nodes für primäre Anwendungsprozesse zu verwenden.
- Wenn Sie Node Labels verwenden, kann die Gesamtzahl der laufenden Einheiten im Cluster vorübergehend die in Ihrer verwalteten Skalierungsrichtlinie festgelegte maximale Rechenleistung überschreiten, während Amazon EMR einige Ihrer Instances außer Betrieb nimmt. Die Gesamtzahl der angeforderten Einheiten bleibt immer auf oder unter der maximalen Rechenleistung Ihrer Richtlinie.
- Managed Scaling unterstützt nur die Knotenbezeichnungen ON_DEMAND und SPOT oder CORE undTASK. Benutzerdefinierte Knotenbezeichnungen werden nicht unterstützt.

- Amazon EMR erstellt Knotenbezeichnungen, wenn der Cluster erstellt und Ressourcen bereitgestellt werden. Amazon EMR unterstützt das Hinzufügen von Knotenbezeichnungen bei der Neukonfiguration des Clusters nicht. Sie können die Knotenbezeichnungen auch nicht ändern, wenn Sie die verwaltete Skalierung nach dem Start des Clusters konfigurieren.
- Bei der verwalteten Skalierung werden Kern- und Taskknoten unabhängig voneinander auf der Grundlage des Anwendungsprozesses und der Nachfrage der Executoren skaliert. Um Probleme mit dem Verlust von HDFS-Daten beim Herunterfahren der Kerne zu vermeiden, sollten Sie sich an die Standardpraxis für Kernknoten halten. Weitere Informationen zu bewährten Methoden für Kernknoten und HDFS-Replikation finden Sie unter [Überlegungen und bewährte Methoden](#).
- Sie können nicht sowohl den Anwendungsprozess als auch die Executoren nur auf dem Knoten `core` oder dem Knoten platzieren. `ON_DEMAND` Wenn Sie sowohl den Anwendungsprozess als auch die Executoren auf einem der Knoten hinzufügen möchten, verwenden Sie die Konfiguration nicht. `yarn.node-labels.am.default-node-label-expression`

Um beispielsweise sowohl den Anwendungsprozess als auch die Executoren in `ON_DEMAND` Knoten zu platzieren, setzen Sie `max compute` auf den Wert `Maximum` im Knoten. `ON_DEMAND` Entfernen Sie außerdem die Konfiguration `yarn.node-labels.am.default-node-label-expression`.

Um sowohl den Anwendungsprozess als auch Executoren auf den `core` Knoten hinzuzufügen, entfernen Sie die `yarn.node-labels.am.default-node-label-expression` Konfiguration.

- Wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden, legen Sie die Eigenschaft fest, `yarn.scheduler.capacity.maximum-am-resource-percent: 1` wenn Sie mehrere Anwendungen parallel ausführen möchten. Dadurch wird sichergestellt, dass Ihre Anwendungsprozesse die verfügbaren `CORE ON_DEMAND OR`-Knoten vollständig nutzen.
- Wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden, legen Sie für `yarn.resourcemanager.decommissioning.timeout` die Eigenschaft einen Wert fest, der länger ist als die am längsten laufende Anwendung in Ihrem Cluster. Dadurch wird die Wahrscheinlichkeit verringert, dass Amazon EMR Managed Scaling Ihre Anwendungen für die Wiederinbetriebnahme oder Knoten neu `CORE` planen muss. `ON_DEMAND`
- Um das Risiko von Anwendungsausfällen aufgrund von Shuffle-Datenverlusten zu verringern, sammelt Amazon EMR Metriken aus dem Cluster, um Knoten zu ermitteln, auf denen transiente Shuffle-Daten aus der aktuellen und vorherigen Phase vorhanden sind. In seltenen Fällen können Metriken weiterhin veraltete Daten für Anwendungen melden, die bereits abgeschlossen oder beendet wurden. Dies kann sich auf die rechtzeitige Reduzierung der Instanzen in Ihrem Cluster

auswirken. Für Cluster mit einer großen Menge an Shuffle-Daten sollten Sie die Verwendung von EMR-Versionen 6.13 und höher in Betracht ziehen.

Feature-Verlauf

In dieser Tabelle sind Aktualisierungen zur Funktion Amazon EMR Managed Scaling aufgeführt.

Datum der Veröffentlichung	Funktion	Versionen für Amazon EMR
20. November 2024	Managed Scaling ist in den Regionen <code>il-central-1</code> Israel (Tel Aviv), <code>me-central-1</code> Naher Osten (VAE) und <code>ap-northeast-3</code> Asien-Pazifik (Osaka) verfügbar.	5.30.0 und 6.1.0 und höher
15. November 2024	Managed Scaling ist in der Region <code>eu-central-2</code> Europa (Zürich) verfügbar.	5.30.0 und 6.1.0 und höher
20. August 2024	Node-Labels sind jetzt in Managed Scaling verfügbar, sodass Sie Ihre Instances nach Markt- oder Knotentyp kennzeichnen können, um die automatische Skalierung zu verbessern.	7.2.0 und höher
31. März 2024	Managed Scaling ist in der Region <code>ap-south-2</code> Asien-Pazifik (Hyderabad) verfügbar.	6.14.0 und höher
13. Februar 2024	Managed Scaling ist in der Region <code>eu-south-2</code> Europa (Spanien) verfügbar.	6.14.0 und höher
10. Oktober 2023	Managed Scaling ist in der <code>ap-southeast-3</code> -Region	6.14.0 und höher

Datum der Veröffentlichung	Funktion	Versionen für Amazon EMR
	Asien-Pazifik (Jakarta) verfügbar.	
28. Juli 2023	Verbesserte verwaltete Skalierung, um beim hochskalieren zu einer anderen Aufgaben-Instance-Gruppe zu wechseln, wenn es bei Amazon EMR beim hochskalieren mit der aktuellen Instance-Gruppe zu Verzögerungen kommt.	5.34.0 und höher, 6.4.0 und höher
16. Juni 2023	Verbesserte verwaltete Skalierung, sodass erkannt wird, auf welchen Knoten der Application Master ausgeführt wird, sodass diese Knoten nicht herunterskaliert werden. Weitere Informationen finden Sie unter Grundlegendes zur Amazon EMR-Knotenzuweisungsstrategie und -Szenarien .	5.34.0 und höher, 6.4.0 und höher

Datum der Veröffentlichung	Funktion	Versionen für Amazon EMR
21. März 2022	Spark Shuffle Data Awareness wurde hinzugefügt, das beim Herunterskalieren von Clustern verwendet wird. Für Amazon-EMR-Cluster mit Apache Spark und aktiviertem verwaltetem Skalierungsfeature überwacht Amazon EMR kontinuierlich Spark-Executoren und Zwischenspeicherorte für Shuffle-Daten. Anhand dieser Informationen skaliert Amazon EMR nur ungenutzte Instances herunter, die keine aktiv genutzten Shuffle-Daten enthalten. Dadurch wird eine Neuberechnung verloren gegangener Shuffle-Daten verhindert, was zur Senkung der Kosten und zur Verbesserung der Arbeitsleistung beiträgt. Weitere Informationen finden Sie unter im Spark-Programmierhandbuch .	5.34.0 und höher, 6.4.0 und höher

Verwaltete Skalierung für Amazon EMR konfigurieren

In den folgenden Abschnitten wird erklärt, wie Sie einen EMR-Cluster starten, der verwaltete Skalierung mit dem AWS Management Console, dem AWS SDK für Java, dem oder dem AWS Command Line Interface verwendet.

Themen

- [Verwenden Sie den AWS Management Console, um die verwaltete Skalierung zu konfigurieren](#)

- [Verwenden Sie den AWS CLI , um die verwaltete Skalierung zu konfigurieren](#)
- [Wird verwendet AWS SDK für Java , um verwaltete Skalierung zu konfigurieren](#)

Verwenden Sie den AWS Management Console , um die verwaltete Skalierung zu konfigurieren

Sie können die Amazon-EMR-Konsole verwenden, um die verwaltete Skalierung zu konfigurieren, wenn Sie einen Cluster erstellen, oder um eine verwaltete Skalierungsrichtlinie für einen laufenden Cluster zu ändern.

Console

Um die verwaltete Skalierung zu konfigurieren, wenn Sie einen Cluster mit der Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie eine Amazon-EMR-Version emr-5.30.0 oder höher, außer Version emr-6.0.0.
4. Wählen Sie unter Option Clusterskalierung und -Bereitstellungsoption EMR-verwaltete Skalierung verwenden aus. Geben Sie das Minimum – und Maximum von Instances, die maximale Anzahl an Core-Knoten-Instances und die maximale Anzahl von On-Demand-Instances an.
5. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
6. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

Um verwaltete Skalierung auf einem vorhandenen Cluster mit der Konsole zu konfigurieren

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten.
3. Suchen Sie auf der Registerkarte Instances der Cluster-Detailseite den Abschnitt Instance-Gruppen-Einstellungen. Geben Sie im Abschnitt Clusterskalierung bearbeiten neue Werte für die Minimum- und Maximum-Anzahl von Instances und das On-Demand-Limit an.

Verwenden Sie den AWS CLI , um die verwaltete Skalierung zu konfigurieren

Sie können AWS CLI Befehle für Amazon EMR verwenden, um die verwaltete Skalierung zu konfigurieren, wenn Sie einen Cluster erstellen. Sie können eine Kurzschreibweise mit der passenden JSON-Konfiguration in den entsprechenden Befehlen oder eine Referenzdatei mit der JSON-Konfiguration verwenden. Sie können auch eine Richtlinie für verwaltete Skalierung auf einen vorhandenen Cluster anwenden und eine zuvor angewendete Richtlinie für verwaltete Skalierung entfernen. Darüber hinaus können Sie Details einer Skalierungsrichtlinien-Konfiguration aus einem aktuell ausgeführten Cluster abrufen.

Aktivieren der verwalteten Skalierung während des Clusterstarts

Sie können die verwaltete Skalierung während des Clusterstarts aktivieren, wie im folgenden Beispiel veranschaulicht wird.

```
aws emr create-cluster \  
  --service-role EMR_DefaultRole \  
  --release-label emr-7.9.0 \  
  --name EMR_Managed_Scaling_Enabled_Cluster \  
  --applications Name=Spark Name=Hbase \  
  --ec2-attributes KeyName=keyName,InstanceProfile=EMR_EC2_DefaultRole \  
  --instance-groups InstanceType=m4.xlarge,InstanceGroupType=MASTER,InstanceCount=1  
  InstanceType=m4.xlarge,InstanceGroupType=CORE,InstanceCount=2 \  
  --region us-east-1 \  
  --managed-scaling-policy  
  ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

Sie können bei Verwendung der managed-scaling-policy Option -- auch eine verwaltete Richtlinienkonfiguration angeben. create-cluster

Anwenden einer Richtlinie für verwaltete Skalierung auf einen vorhandenen Cluster

Sie können eine Richtlinie für verwaltete Skalierung auf einen vorhandenen Cluster anwenden, wie im folgenden Beispiel veranschaulicht wird.

```
aws emr put-managed-scaling-policy  
  --cluster-id j-123456  
  --managed-scaling-policy ComputeLimits='{MinimumCapacityUnits=1,  
  MaximumCapacityUnits=10, MaximumOnDemandCapacityUnits=10, UnitType=Instances}'
```

Sie können eine Richtlinie für verwaltete Skalierung auch auf einen vorhandenen Cluster anwenden, indem Sie den Befehl `aws emr put-managed-scaling-policy` verwenden. Im folgenden Beispiel wird ein Verweis auf eine JSON-Datei verwendet, `managedscaleconfig.json`, die die Konfiguration der Richtlinie für verwaltete Skalierung angibt.

```
aws emr put-managed-scaling-policy --cluster-id j-123456 --managed-scaling-policy  
file:///./managedscaleconfig.json
```

Das folgende Beispiel zeigt den Inhalt der Datei `managedscaleconfig.json`, in der die Richtlinie für verwaltete Skalierung definiert wird.

```
{  
  "ComputeLimits": {  
    "UnitType": "Instances",  
    "MinimumCapacityUnits": 1,  
    "MaximumCapacityUnits": 10,  
    "MaximumOnDemandCapacityUnits": 10  
  }  
}
```

Abrufen einer Richtlinienkonfiguration für verwaltete Skalierung

Der Befehl `GetManagedScalingPolicy` ruft die Richtlinienkonfiguration ab. Mit dem folgenden Befehl wird beispielsweise die Konfiguration für den Cluster mit der Cluster-ID `j-123456` abgerufen.

```
aws emr get-managed-scaling-policy --cluster-id j-123456
```

Der Befehl generiert die folgende Beispielausgabe:

```
{  
  "ManagedScalingPolicy": {  
    "ComputeLimits": {  
      "MinimumCapacityUnits": 1,  
      "MaximumOnDemandCapacityUnits": 10,  
      "MaximumCapacityUnits": 10,  
      "UnitType": "Instances"  
    }  
  }  
}
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Entfernen der Richtlinie für verwaltete Skalierung

Mit dem Befehl `RemoveManagedScalingPolicy` wird die Richtlinienkonfiguration entfernt. Mit dem folgenden Befehl wird beispielsweise die Konfiguration für den Cluster mit der Cluster-ID `j-123456` entfernt.

```
aws emr remove-managed-scaling-policy --cluster-id j-123456
```

Wird verwendet AWS SDK für Java , um verwaltete Skalierung zu konfigurieren

Der folgende Programmausschnitt zeigt, wie die verwaltete Skalierung mit dem AWS SDK für Java konfiguriert wird:

```
package com.amazonaws.emr.sample;

import java.util.ArrayList;
import java.util.List;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.Application;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimits;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimitsUnitType;
import com.amazonaws.services.elasticmapreduce.model.InstanceGroupConfig;
import com.amazonaws.services.elasticmapreduce.model.JobFlowInstancesConfig;
import com.amazonaws.services.elasticmapreduce.model.ManagedScalingPolicy;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowRequest;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowResult;

public class CreateClusterWithManagedScalingWithIG {

    public static void main(String[] args) {
        AWSCredentials credentialsFromProfile = getCredentials("AWS-Profile-Name-Here");

        /**
```

```

    * Create an Amazon EMR client with the credentials and region specified in order to
    create the cluster
    */
    AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
        .withCredentials(new AWSStaticCredentialsProvider(credentialsFromProfile))
        .withRegion(Regions.US_EAST_1)
        .build();

    /**
     * Create Instance Groups - Primary, Core, Task
     */
    InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
        .withInstanceCount(1)
        .withInstanceRole("MASTER")
        .withInstanceType("m4.large")
        .withMarket("ON_DEMAND");

    InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
        .withInstanceCount(4)
        .withInstanceRole("CORE")
        .withInstanceType("m4.large")
        .withMarket("ON_DEMAND");

    InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
        .withInstanceCount(5)
        .withInstanceRole("TASK")
        .withInstanceType("m4.large")
        .withMarket("ON_DEMAND");

    List<InstanceGroupConfig> igConfigs = new ArrayList<>();
    igConfigs.add(instanceGroupConfigMaster);
    igConfigs.add(instanceGroupConfigCore);
    igConfigs.add(instanceGroupConfigTask);

    /**
     * specify applications to be installed and configured when Amazon EMR creates
    the cluster
     */
    Application hive = new Application().withName("Hive");
    Application spark = new Application().withName("Spark");
    Application ganglia = new Application().withName("Ganglia");
    Application zeppelin = new Application().withName("Zeppelin");

    /**

```

```

* Managed Scaling Configuration -
    * Using UnitType=Instances for clusters composed of instance groups
*
    * Other options are:
    * UnitType = VCPU ( for clusters composed of instance groups)
    * UnitType = InstanceFleetUnits ( for clusters composed of instance fleets)
    **/
ComputeLimits computeLimits = new ComputeLimits()
    .withMinimumCapacityUnits(1)
    .withMaximumCapacityUnits(20)
    .withUnitType(ComputeLimitsUnitType.Instances);

ManagedScalingPolicy managedScalingPolicy = new ManagedScalingPolicy();
managedScalingPolicy.setComputeLimits(computeLimits);

// create the cluster with a managed scaling policy
RunJobFlowRequest request = new RunJobFlowRequest()
    .withName("EMR_Managed_Scaling_TestCluster")
    .withReleaseLabel("emr-7.9.0")           // Specifies the version label for
the Amazon EMR release; we recommend the latest release
    .withApplications(hive,spark,ganglia,zeppelin)
    .withLogUri("s3://path/to/my/emr/logs") // A URI in S3 for log files is
required when debugging is enabled.
    .withServiceRole("EMR_DefaultRole")      // If you use a custom IAM service
role, replace the default role with the custom role.
    .withJobFlowRole("EMR_EC2_DefaultRole") // If you use a custom Amazon EMR
role for EC2 instance profile, replace the default role with the custom Amazon EMR
role.
    .withInstances(new JobFlowInstancesConfig().withInstanceGroups(igConfigs)
        .withEc2SubnetId("subnet-123456789012345")
        .withEc2KeyName("my-ec2-key-name")
        .withKeepJobFlowAliveWhenNoSteps(true))
    .withManagedScalingPolicy(managedScalingPolicy);
RunJobFlowResult result = emr.runJobFlow(request);

System.out.println("The cluster ID is " + result.toString());
}

public static AWSCredentials getCredentials(String profileName) {
    // specifies any named profile in .aws/credentials as the credentials provider
    try {
        return new ProfileCredentialsProvider("AWS-Profile-Name-Here")
            .getCredentials();
    } catch (Exception e) {

```



```
        throw new AmazonClientException(
            "Cannot load credentials from .aws/credentials file. " +
            "Make sure that the credentials file exists and that the profile\n        name is defined within it.",
            e);
    }
}

public CreateClusterWithManagedScalingWithIG() { }
}
```

Erweiterte Skalierung für Amazon EMR

Ab Amazon EMR auf EC2 Version 7.0 können Sie Advanced Scaling nutzen, um die Ressourcennutzung Ihres Clusters zu kontrollieren. Advanced Scaling führt eine Nutzungs- und Leistungsskala ein, mit der Sie Ihre Ressourcennutzung und Ihr Leistungsniveau an Ihre Geschäftsanforderungen anpassen können. Der von Ihnen festgelegte Wert bestimmt, ob Ihr Cluster stärker auf Ressourcenschonung oder auf Skalierung zur Bewältigung service-level-agreement (SLA) sensibler Workloads ausgerichtet ist, bei denen eine schnelle Fertigstellung entscheidend ist. Wenn der Skalierungswert angepasst wird, interpretiert die verwaltete Skalierung Ihre Absicht und skaliert intelligent, um die Ressourcen zu optimieren. Weitere Informationen zur verwalteten Skalierung finden [Sie unter Managed Scaling für Amazon EMR konfigurieren](#).

Erweiterte Skalierungseinstellungen

Der Wert, den Sie für Advanced Scaling festgelegt haben, optimiert Ihren Cluster an Ihre Anforderungen. Die Werte liegen im Bereich von 1 bis 100. Mögliche Werte sind 1, 25, 50, 75 und 100. Wenn Sie den Index auf andere Werte als diese festlegen, führt dies zu einem Validierungsfehler.

Skalierungswerte entsprechen Strategien zur Ressourcennutzung. In der folgenden Liste werden mehrere davon definiert:

- **Optimierte Nutzung [1]** — Diese Einstellung verhindert eine übermäßige Bereitstellung von Ressourcen. Verwenden Sie einen niedrigen Wert, wenn Sie die Kosten niedrig halten und einer effizienten Ressourcennutzung Priorität einräumen möchten. Dies führt dazu, dass der Cluster weniger aggressiv skaliert wird. Dies eignet sich gut für den Anwendungsfall, in dem regelmäßig Arbeitslastspitzen auftreten und Sie nicht möchten, dass die Ressourcen zu schnell ansteigen.
- **Ausgewogen [50]** — Dadurch wird ein ausgewogenes Verhältnis zwischen Ressourcennutzung und Arbeitsleistung erreicht. Diese Einstellung eignet sich für konstante Workloads, bei denen die

meisten Phasen über eine stabile Laufzeit verfügen. Sie eignet sich auch für Workloads mit einer Mischung aus Phasen mit kurzer und langer Laufzeit. Wir empfehlen, mit dieser Einstellung zu beginnen, wenn Sie sich nicht sicher sind, welche Sie wählen sollen.

- **Leistungsoptimiert [100]** — Bei dieser Strategie wird Leistung priorisiert. Der Cluster wird aggressiv skaliert, um sicherzustellen, dass Jobs schnell abgeschlossen werden und die Leistungsziele erreicht werden. Performance Optimized eignet sich für service-level-agreement (SLA-) sensible Workloads, bei denen eine schnelle Laufzeit entscheidend ist.

Note

Die verfügbaren Zwischenwerte bieten einen Mittelweg zwischen den Strategien zur Feinabstimmung des Advanced Scaling-Verhaltens Ihres Clusters.

Vorteile der erweiterten Skalierung

Da Ihre Umgebung und Ihre Anforderungen unterschiedlich sind, z. B. sich ändernde Datenmengen, Anpassungen der Kostenziele und SLA-Implementierungen, kann die Clusterskalierung Ihnen helfen, Ihre Clusterkonfiguration so anzupassen, dass Sie Ihre Ziele erreichen. Zu den wichtigsten Vorteilen gehören:

- **Verbesserte detaillierte Steuerung** — Mit der Einführung der Einstellung für die Auslastung und Leistung können Sie das Skalierungsverhalten Ihres Clusters ganz einfach an Ihre Anforderungen anpassen. Sie können je nach Ihren Nutzungsmustern nach oben skalieren, um den Bedarf an Rechenressourcen zu decken, oder nach unten skalieren, um Ressourcen zu sparen.
- **Verbesserte Kostenoptimierung** — Sie können je nach Bedarf einen niedrigen Nutzungswert wählen, um Ihre Kostenziele leichter zu erreichen.

Erste Schritte mit der Optimierung

Einrichtung und Konfiguration

Gehen Sie wie folgt vor, um den Leistungsindex festzulegen und Ihre Skalierungsstrategie zu optimieren.

1. Mit dem folgenden Befehl wird ein vorhandener Cluster mit der nutzungsoptimierten [1] Skalierungsstrategie aktualisiert:

```
aws emr put-managed-scaling-policy --cluster-id 'cluster-id' \
--managed-scaling-policy '{
  "ComputeLimits": {
    "UnitType": "Instances",
    "MinimumCapacityUnits": 1,
    "MaximumCapacityUnits": 2,
    "MaximumOnDemandCapacityUnits": 2,
    "MaximumCoreCapacityUnits": 2
  },
  "ScalingStrategy": "ADVANCED",
  "UtilizationPerformanceIndex": "1"
}' \
--region "region-name"
```

Die Attribute `ScalingStrategy` und `UtilizationPerformanceIndex` sind neu und für die Skalierungsoptimierung relevant. Sie können verschiedene Skalierungsstrategien auswählen, indem Sie die entsprechenden Werte (1, 25, 50, 75 und 100) für das `UtilizationPerformanceIndex` Attribut in der Richtlinie für verwaltete Skalierung festlegen.

2. Um zur Standardstrategie für verwaltete Skalierung zurückzukehren, führen Sie den `put-managed-scaling-policy` Befehl aus, ohne die Attribute und einzubeziehen. `ScalingStrategy` `UtilizationPerformanceIndex` (Dies ist optional.) Dieses Beispiel zeigt, wie das geht:

```
aws emr put-managed-scaling-policy \
--cluster-id 'cluster-id' \
--managed-scaling-policy '{"ComputeLimits":
{"UnitType":"Instances","MinimumCapacityUnits":1,"MaximumCapacityUnits":2,"MaximumOnDemandC
\
--region "region-name"
```

Verwendung von Überwachungsmetriken zur Nachverfolgung der Cluster-Auslastung

Ab EMR-Version 7.3.0 veröffentlicht Amazon EMR vier neue Metriken zu Arbeitsspeicher und virtueller CPU. Sie können diese verwenden, um die Cluster-Auslastung über verschiedene Skalierungsstrategien hinweg zu messen. Diese Metriken sind für jeden Anwendungsfall verfügbar, aber Sie können die hier bereitgestellten Details zur Überwachung von Advanced Scaling verwenden.

Zu den verfügbaren hilfreichen Metriken gehören die folgenden:

- **YarnContainersUsedMemoryGBSeconds**— Speichermenge, die von Anwendungen verbraucht wird, die von YARN verwaltet werden.
- **YarnContainersTotalMemoryGBSeconds**— Gesamtspeicherkapazität, die YARN innerhalb des Clusters zugewiesen wurde.
- **YarnNodesUsedVCPUSeconds**— VCPU-Sekunden insgesamt für jede von YARN verwaltete Anwendung.
- **YarnNodesTotalVCPUSeconds**— Aggregierte Gesamt-VCPU-Sekunden für den verbrauchten Speicher, einschließlich des Zeitfensters, in dem Yarn nicht bereit ist.

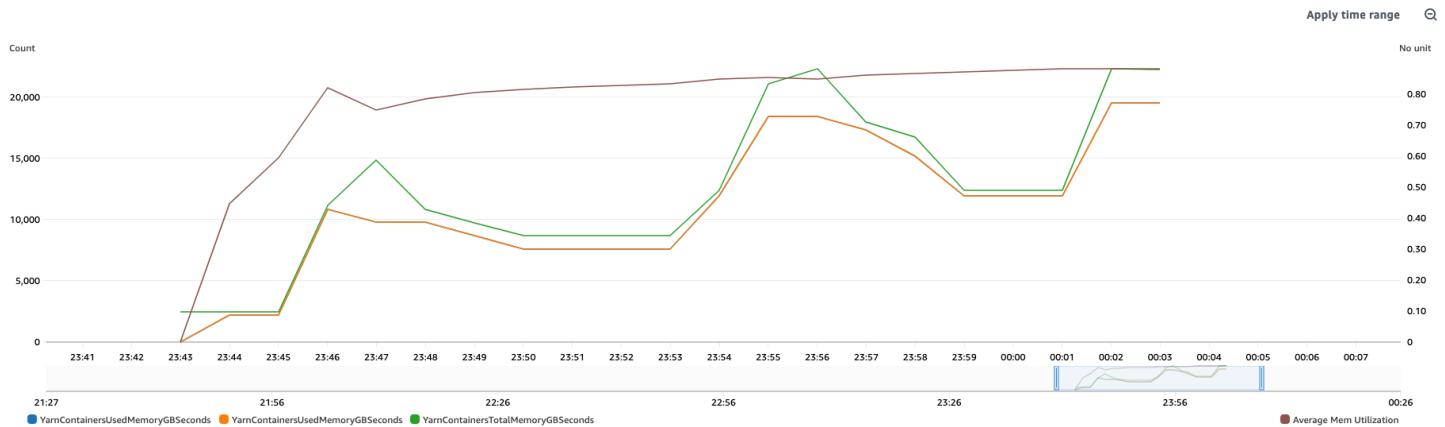
Sie können Ressourcenmetriken mit Amazon CloudWatch Logs Insights analysieren. Zu den Funktionen gehört eine speziell entwickelte Abfragesprache, mit der Sie Kennzahlen extrahieren können, die für die Ressourcennutzung und Skalierung spezifisch sind.

Die folgende Abfrage, die Sie in der Amazon CloudWatch Konsole ausführen können, verwendet metrische Mathematik, um die durchschnittliche Speicherauslastung (e1) zu berechnen, indem die laufende Summe des verbrauchten Speichers (e2) durch die laufende Summe des Gesamtspeichers (e3) dividiert wird:

```
{
  "metrics": [
    [ { "expression": "e2/e3", "label": "Average Mem Utilization", "id": "e1",
      "yAxis": "right" } ],
    [ { "expression": "RUNNING_SUM(m1)", "label": "RunningTotal-
YarnContainersUsedMemoryGBSeconds", "id": "e2", "visible": false } ],
    [ { "expression": "RUNNING_SUM(m2)", "label": "RunningTotal-
YarnContainersTotalMemoryGBSeconds", "id": "e3", "visible": false } ],
    [ "AWS_EMR_ManagedResize", "YarnContainersUsedMemoryGBSeconds", "ACCOUNT_ID",
      "793684541905", "COMPONENT", "ManagerService", "JOB_FLOW_ID", "cluster-id", { "id":
      "m1", "label": "YarnContainersUsedMemoryGBSeconds" } ],
    [ ".", "YarnContainersTotalMemoryGBSeconds", ".", ".", ".", ".", ".", ".",
      { "id": "m2", "label": "YarnContainersTotalMemoryGBSeconds" } ]
  ],
  "view": "timeSeries",
  "stacked": false,
  "region": "region",
  "period": 60,
  "stat": "Sum",
  "title": "Memory Utilization"
}
```

Um Protokolle abzufragen, können Sie CloudWatch in der AWS Konsole auswählen. Weitere Informationen zum Schreiben von Abfragen für CloudWatch finden Sie unter [Analysieren von Protokolldaten mit CloudWatch Logs Insights](#) im Amazon CloudWatch Logs-Benutzerhandbuch.

Die folgende Abbildung zeigt diese Metriken für einen Beispielcluster:



Überlegungen und Einschränkungen

- Die Effektivität von Skalierungsstrategien kann je nach Ihren individuellen Workload-Merkmalen und der Cluster-Konfiguration variieren. Wir empfehlen Ihnen, mit der Skalierungseinstellung zu experimentieren, um einen optimalen Indexwert für Ihren Anwendungsfall zu ermitteln.
- Amazon EMR Advanced Scaling eignet sich besonders gut für Batch-Workloads. Für SQL/Data-Warehousing- und Streaming-Workloads empfehlen wir die Verwendung der Standardstrategie für verwaltete Skalierung, um eine optimale Leistung zu erzielen.
- Die leistungsoptimierte Skalierungsstrategie ermöglicht eine schnellere Auftragsausführung, indem hohe Rechenressourcen über einen längeren Zeitraum aufrechterhalten werden als die standardmäßige Strategie für verwaltete Skalierung. Dieser Modus priorisiert die schnelle Skalierung, um den Ressourcenanforderungen gerecht zu werden, was zu einer schnelleren Auftragsabwicklung führt. Dies kann im Vergleich zur Standardstrategie zu höheren Kosten führen.
- In Fällen, in denen der Cluster bereits optimiert und voll ausgelastet ist, bietet die Aktivierung von Advanced Scaling möglicherweise keine zusätzlichen Vorteile. In einigen Situationen kann die Aktivierung von Advanced Scaling zu erhöhten Kosten führen, da die Workloads länger laufen können. In diesen Fällen empfehlen wir, die Standardstrategie für verwaltete Skalierung zu verwenden, um eine optimale Ressourcenzuweisung und Kosteneffizienz zu gewährleisten.
- Im Zusammenhang mit verwalteter Skalierung verlagert sich der Schwerpunkt auf die Ressourcennutzung im Laufe der Ausführungszeit, da die Einstellung von leistungsoptimiert [100] auf auslastungsoptimiert [1] geändert wird. Es ist jedoch wichtig zu beachten, dass die

Ergebnisse je nach Art der Arbeitslast und der Topologie des Clusters variieren können. Um optimale Ergebnisse für Ihren Anwendungsfall sicherzustellen, empfehlen wir dringend, die Skalierungsstrategien mit Ihren Workloads zu testen, um die am besten geeignete Einstellung zu ermitteln.

- Der PerformanceUtilizationIndex akzeptiert nur die folgenden Werte:
 - 1
 - 25
 - 50
 - 75
 - 100

Alle anderen übermittelten Werte führen zu einem Validierungsfehler.

Grundlegendes zur Amazon EMR-Knotenzuweisungsstrategie und -Szenarien

Dieser Abschnitt gibt einen Überblick über die Strategie zur Knotenzuweisung und allgemeine Skalierungsszenarien, die Sie mit Amazon EMR Managed Scaling verwenden können.

Knotenzuweisungsstrategie

Amazon EMR Managed Scaling weist Core- und Aufgabenknoten auf der Grundlage der folgenden Strategien zum hochskalieren und herunterskalieren zu:

Strategie zum hochskalieren

- Für Amazon EMR-Versionen 7.2 und höher fügt die verwaltete Skalierung zunächst Knoten hinzu, die auf Knotenbezeichnungen und der YARN-Eigenschaft für Anwendungsprozesseinschränkungen basieren.
- Für Amazon EMR-Versionen 7.2 und höher gilt: Wenn Sie Node Labels aktiviert und Anwendungsprozesse auf CORE Knoten beschränkt haben, skaliert Amazon EMR Managed Scaling die Kernknoten und Aufgabenknoten hoch, wenn die Nachfrage nach Anwendungsprozessen steigt und die Nachfrage nach Ausführern steigt. Wenn Sie Node-Labels aktiviert und Anwendungsprozesse auf Knoten beschränkt haben, skaliert Managed Scaling entsprechend ON_DEMAND On-Demand-Knoten, wenn die Nachfrage nach Anwendungsprozessen steigt, und skaliert Spot-Knoten, wenn die Nachfrage nach Executoren steigt.
- Wenn Node Labels nicht aktiviert sind, ist die Platzierung von Anwendungsprozessen nicht auf einen Knoten oder Markttyp beschränkt.

- Durch die Verwendung von Node Labels kann Managed Scaling verschiedene Instanzgruppen und Instanzflotten bei derselben Größenänderung hoch- und herunterskalieren. Zum Beispiel in einem Szenario, in dem `instance_group1` ein `ON_DEMAND` Knoten und ein `SPOT` Knoten `instance_group2` vorhanden sind und die Knotenbezeichnungen aktiviert sind und Anwendungsprozesse auf Knoten mit der `ON_DEMAND` Bezeichnung beschränkt sind. Die verwaltete Skalierung wird herunterskaliert `instance_group1` und hochskaliert, `instance_group2` wenn die Nachfrage nach Anwendungsprozessen sinkt und die Nachfrage nach Ausführern steigt.
- Wenn Amazon EMR bei der Skalierung mit der aktuellen Instance-Gruppe verzögert wird, wechseln Cluster, die verwaltete Skalierung verwenden, automatisch zu einer anderen Task-Instance-Gruppe.
- Wenn der `MaximumCoreCapacityUnits`-Parameter festgelegt ist, skaliert Amazon EMR die Core-Knoten, bis die Kerneinheiten den maximal zulässigen Grenzwert erreichen. Die gesamte verbleibende Kapazität wird den Aufgabenknoten hinzugefügt.
- Wenn der `MaximumOnDemandCapacityUnits`-Parameter festgelegt ist, skaliert Amazon EMR den Cluster mithilfe der On-Demand-Instances, bis die On-Demand-Einheiten den maximal zulässigen Grenzwert erreichen. Die gesamte verbleibende Kapazität wird mithilfe von Spot Instances hinzugefügt.
- Wenn sowohl `MaximumCoreCapacityUnits` als auch der `MaximumOnDemandCapacityUnits` Parameter festgelegt sind, berücksichtigt Amazon EMR bei der Skalierung beide Grenzwerte.

Wenn `MaximumCoreCapacityUnits` beispielsweise kleiner als `MaximumOnDemandCapacityUnits` ist, skaliert Amazon EMR zunächst die Core-Knoten, bis die Kernkapazitätsgrenze erreicht ist. Für die verbleibende Kapazität verwendet Amazon EMR zunächst On-Demand-Instances, um Aufgabenknoten zu skalieren, bis das On-Demand-Limit erreicht ist, und verwendet dann Spot Instances für Aufgabenknoten.

Strategie zum herunterskalieren

- Ähnlich wie bei der Scale-up-Strategie entfernt Amazon EMR Knoten, die auf Knotenbezeichnungen basieren. Weitere Informationen zu Knotenbezeichnungen finden Sie unter Grundlegendes [zu Knotentypen: Primär-, Kern- und Aufgabenknoten](#).
- Wenn Sie Node Labels nicht aktiviert haben, entfernt Managed Scaling Task-Knoten und anschließend Core-Knoten, bis die gewünschte Scale-Down-Zielkapazität erreicht ist. Bei der verwalteten Skalierung wird der Cluster niemals unter die in der Richtlinie für verwaltete Skalierung angegebenen Mindestbeschränkungen herunterskaliert.

- Amazon EMR-Versionen 5.34.0 und höher sowie Amazon EMR-Versionen 6.4.0 und höher unterstützen Spark Shuffle Data Awareness, wodurch verhindert wird, dass eine Instance herunterskaliert wird, während Managed Scaling vorhandene Shuffle-Daten erkennt. [Weitere Informationen zu Shuffle-Vorgängen finden Sie im Spark-Programmierhandbuch.](#) Managed Scaling bemüht sich nach besten Kräften, das Herunterskalieren von Knoten mit Shuffle-Daten aus der aktuellen und vorherigen Phase einer aktiven Spark-Anwendung zu verhindern, und zwar bis zu einem Maximum von 30 Minuten. Dies trägt dazu bei, den unbeabsichtigten Verlust von Shuffle-Daten zu minimieren, sodass keine erneuten Versuche und die Neuberechnung von Zwischendaten erforderlich sind. Die Verhinderung des Verlusts von Shuffle-Daten kann jedoch nicht garantiert werden. Für einen garantierten Schutz empfehlen wir Shuffle Awareness auf Clustern mit Release-Label 7.4 oder höher. Im Folgenden erfahren Sie, wie Sie den garantierten Shuffle-Schutz einrichten.
- Bei der verwalteten Skalierung werden zuerst Task-Knoten und dann Core-Knoten entfernt, bis die gewünschte Scale-Down-Zielkapazität erreicht ist. Der Cluster wird niemals unter die in der Richtlinie für verwaltete Skalierung angegebenen Mindestbeschränkungen skaliert.
- Bei Clustern, die mit Amazon EMR 5.x-Versionen 5.34.0 und höher und 6.x-Versionen 6.4.0 und höher gestartet werden, skaliert Amazon EMR Managed Scaling keine Knoten, auf denen `ApplicationMaster` für Apache Spark ausgeführt wird. Dadurch werden Fehlschläge und Wiederholungen von Aufträgen minimiert, was zur Verbesserung der Auftragsleistung und zur Senkung der Kosten beiträgt. Um zu überprüfen, welche Knoten in Ihrem Cluster `ApplicationMaster` ausführen, besuchen Sie den Spark History Server und filtern Sie auf der Registerkarte Executors Ihrer Spark-Anwendungs-ID nach dem Treiber.
- Die intelligente Skalierung mit EMR Managed Scaling minimiert zwar den Verlust von Shuffle-Daten für Spark, es kann jedoch vorkommen, dass vorübergehende Shuffle-Daten während eines Scale-Down möglicherweise nicht geschützt sind. Um die Ausfallsicherheit von Shuffle-Daten beim Herunterskalieren zu erhöhen, empfehlen wir, Graceful Decommissioning for Shuffle Data in YARN zu aktivieren. Wenn Graceful Decommissioning for Shuffle Data in YARN aktiviert ist, gehen Knoten, die für das Herunterskalieren ausgewählt wurden und über Shuffle-Daten verfügen, in den Status Außerbetriebnahme über und stellen weiterhin Shuffle-Dateien bereit. Das YARN ResourceManager wartet, bis die Knoten melden, dass keine Shuffle-Dateien vorhanden sind, bevor es die Knoten aus dem Cluster entfernt.
- Amazon EMR Version 6.11.0 und höher unterstützt Yarn-basierte Graceful Decommissioning für Hive-Shuffle-Daten sowohl für den Tez- als auch für den Shuffle-Handler. MapReduce

- Aktivieren Sie die automatische true Außerbetriebnahme für Shuffle Data, indem Sie auf `yarn.resourcemanager.decommissioning-nodes-watcher.wait-for-shuffle-data` einstellen.
- Amazon EMR, Version 7.4.0 und höher, unterstützt Yarn-basierte Graceful Decommissioning für Spark-Shuffle-Daten, wenn der externe Shuffle-Dienst aktiviert ist (standardmäßig aktiviert, wenn EMR aktiviert ist). EC2
- Das Standardverhalten des externen Spark-Shuffle-Dienstes bei der Ausführung von Spark auf Yarn besteht darin, dass Yarn die Shuffle-Dateien der Anwendung beim Beenden der Anwendung entfernt NodeManager . Dies kann sich auf die Geschwindigkeit der Außerbetriebnahme von Knoten und die Rechenauslastung auswirken. Bei Anwendungen mit langer Laufzeit sollten Sie erwägen, `spark.shuffle.service.removeShuffle` die Einstellung `true` auf so einzustellen, dass nicht mehr verwendete Shuffle-Dateien entfernt werden, um Knoten ohne aktive Shuffle-Daten schneller außer Betrieb zu nehmen.
- Wenn entweder das `yarn.nodemanager.shuffledata-monitor.interval-ms` Kennzeichen oder das gegenüber den Standardwerten geändert `spark.dynamicAllocation.executorIdleTimeout` wurde, stellen Sie sicher, dass der Zustand `spark.dynamicAllocation.executorIdleTimeout > yarn.nodemanager.shuffledata-monitor.interval-ms` erhalten bleibt, `true` indem Sie das erforderliche Kennzeichen aktualisieren.

Wenn der Cluster nicht ausgelastet ist, storniert Amazon EMR das Hinzufügen neuer Instances aus einer früheren Evaluierung und führt Herunterskalierungsvorgänge durch. Wenn der Cluster stark ausgelastet ist, bricht Amazon EMR das Entfernen von Instances ab und führt Hochskalierungsvorgänge durch.

Überlegungen zur Knotenzuweisung

Wir empfehlen, die On-Demand-Kaufoption für Core-Knoten zu verwenden, um HDFS-Datenverlust im Falle einer Spot-Rückforderung zu vermeiden. Sie können die Spot-Kaufoption für Aufgabenknoten verwenden, um die Kosten zu senken und die Auftragsausführung zu beschleunigen, wenn mehr Spot Instances zu Aufgabenknoten hinzugefügt werden.

Knotenzuweisungsszenarien

Sie können je nach Bedarf verschiedene Skalierungsszenarien erstellen, indem Sie die Core-Knotenparameter Maximum, Minimum, On-Demand-Limit und Maximum in unterschiedlichen Kombinationen einrichten.

Szenario 1: Nur Core-Knoten skalieren

Um nur Core-Knoten zu skalieren, müssen die verwalteten Skalierungsparameter die folgenden Anforderungen erfüllen:

- Das On-Demand-Limit entspricht der maximalen Grenze.
- Der maximale Core-Knoten entspricht der maximalen Grenze.

Wenn das On-Demand-Limit und die maximale Anzahl an Core-Knoten nicht angegeben sind, verwenden beide Parameter standardmäßig die maximale Grenze.

Dieses Szenario ist nicht anwendbar, wenn Sie Managed Scaling mit Node Labels verwenden und Ihre Anwendungsprozesse darauf beschränken, nur auf CORE Knoten ausgeführt zu werden, da Managed Scaling Task-Knoten skaliert, um den Anforderungen der Executoren gerecht zu werden.

Das folgende Beispiele zeigt das Szenario der ausschließlichen Skalierung von Core-Knoten.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen Core: 1 On-Demand Aufgabe: 1 On-Demand- und 1 Spot	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 20	Skalieren Sie mithilfe des On-Demand-Typs zwischen 1 und 20 Instances oder Instance-Flotteneinheiten auf Core-Knoten. Keine Skalierung auf Aufgabenknoten.
Instance-Flotten Core: 1 On-Demand Aufgabe: 1 On-Demand- und 1 Spot	UnitType: InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20	Wenn Sie Managed Scaling mit Node Labels verwenden und

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
	MaximumCoreCapacityUnits : 20	Ihre Anwendung prozesse auf ON_DEMAND Knoten beschrän- ken, skaliert der Cluster je nach Art der Nachfrage 1 bis 20 Instanzen oder Instanzfl otteneinh eiten auf CORE Knoten, die den Spot Typ On- Demand oder verwenden.

Szenario 2: Nur Aufgabenknoten skalieren

Um nur Aufgabenknoten zu skalieren, müssen die verwalteten Skalierungsparameter die folgenden Anforderungen erfüllen:

- Der maximale Core-Knoten muss der Mindestgrenze entsprechen.

Das folgende Beispiele zeigt das Szenario der ausschließlichen Skalierung von Aufgabenknoten.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen Core: 2 On-Demand	UnitType: Instances MinimumCapacityUnits : 2	Halten Sie die Anzahl der Core- Knoten konstant

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Aufgabe: 1 Spot	MaximumCapacityUnits : 20 MaximumCoreCapacityUnits : 2	bei 2 und skalieren Sie nur Aufgabenknoten zwischen 0 und 18 Instances oder Instance-Flotteneinheiten. Die Kapazität zwischen Mindest- und Höchstgrenzen wird nur den Aufgabenknoten hinzugefügt.
Instance-Flotten Core: 2 On-Demand Aufgabe: 1 Spot	UnitType: InstanceFleetUnits MinimumCapacityUnits : 2 MaximumCapacityUnits : 20 MaximumCoreCapacityUnits : 2	Wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden und Ihre Anwendungsprozesse auf ON_DEMAND-Knoten beschränken, hält der Cluster die Anzahl der Kernknoten konstant bei 2 und skaliert nur Taskknoten zwischen 0 und 18 Instances

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
		oder Instance-Flotteneinheiten, die den Spot Typ On-demand oder verwenden, je nach Art der Nachfrage.

Szenario 4: Nur On-Demand-Instance im Cluster

Um nur über On-Demand-Instances zu verfügen, müssen Ihr Cluster und die verwalteten Skalierungsparameter die folgende Anforderung erfüllen:

- Das On-Demand-Limit entspricht der maximalen Grenze.

Wenn das On-Demand-Limit nicht angegeben ist, entspricht der Parameterwert standardmäßig der Höchstgrenze. Der Standardwert gibt an, dass Amazon EMR nur On-Demand-Instances skaliert.

Wenn die maximale Anzahl an Core-Knoten kleiner als die maximale Grenze ist, kann der Parameter „Maximaler Core-Knoten“ verwendet werden, um die Kapazitätszuweisung zwischen Core- und Aufgabenknoten aufzuteilen.

Um dieses Szenario in einem Cluster zu aktivieren, der aus Instance-Gruppen besteht, müssen alle Knotengruppen im Cluster bei der Erstkonfiguration den Markttyp On-Demand verwenden.

Dieses Szenario ist nicht anwendbar, wenn Sie Managed Scaling mit Node-Labels verwenden und Ihre Anwendungsprozesse so beschränken, dass sie nur auf Knoten ausgeführt werden, da die verwaltete Skalierung die ON_DEMAND Knoten skaliert, um den Anforderungen der Executoren gerecht zu Spot werden.

Die folgenden Beispiele veranschaulichen das Szenario, in dem On-Demand-Instances im gesamten Cluster vorhanden sind.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen Core: 1 On-Demand Aufgabe: 1 On-Demand	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 12	Skalieren Sie mithilfe des On-Demand-Typs zwischen 1 und 12 Instances oder Instance-Flotteneinheiten auf Core-Knoten. Skalieren Sie die verbleibende Kapazität mithilfe von On-Demand-Funktion auf Aufgabenknoten. Keine Skalierung mit Spot Instances.
Instance-Flotten Core: 1 On-Demand Aufgabe: 1 On-Demand	UnitType: InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 20 MaximumCoreCapacityUnits : 12	Wenn Sie Managed Scaling mit Node Labels verwenden und Ihre Anwendung sprozenesse auf CORE Knoten beschränken, skaliert der Cluster je nach Art der Nachfrage zwischen 1 und 20 Instanzen

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
		oder Instanzflotteneinheiten auf CORE task Knoten oder Knoten, die diesen ON_DEMAND Typ verwenden. Die Skalierung auf Kernknoten wird 12 Instances oder Instance-Flotteneinheiten nicht überschreiten.

Szenario 4: Nur Spot Instances im Cluster

Um nur Spot Instances zu verwenden, müssen die verwalteten Skalierungsparameter die folgenden Anforderungen erfüllen:

- Das On-Demand-Limit ist auf 0 gesetzt.

Wenn die maximale Anzahl an Core-Knoten kleiner als die maximale Grenze ist, kann der Parameter „Maximaler Core-Knoten“ verwendet werden, um die Kapazitätszuweisung zwischen Core- und Aufgabenknoten aufzuteilen.

Um dieses Szenario in einem Cluster zu aktivieren, der aus Instance-Gruppen besteht, muss die Kern-Instance-Gruppe bei der Erstkonfiguration die Spot-Kaufoption verwenden. Wenn in der Aufgaben-Instance-Gruppe keine Spot Instance vorhanden ist, erstellt Amazon EMR Managed Scaling bei Bedarf eine Auftragsgruppe, die Spot Instances verwendet.

Dieses Szenario ist nicht anwendbar, wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden und Ihre Anwendungsprozesse darauf beschränken, nur auf ON_DEMAND Knoten ausgeführt zu werden, da die verwaltete Skalierung die ON_DEMAND Knoten skaliert, um den Anforderungen der Anwendungsprozesse gerecht zu werden.

Die folgenden Beispiele veranschaulichen das Szenario, in dem Spot Instances im gesamten Cluster vorhanden sind.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen Core: 1 Spot Aufgabe: 1 Spot	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 0	Skalieren Sie mithilfe von Spot zwischen 1 und 20 Instances oder Instance-Flotteneinheiten auf Core-Knoten. Keine Skalierung beim On-Demand-Typ.
Instance-Flotten Core: 1 Spot Aufgabe: 1 Spot	UnitType: InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 0	Wenn Sie Managed Scaling mit Node Labels verwenden und Ihre Anwendungsprozesse auf CORE Knoten beschränken, skaliert der Cluster je nach Art des Bedarfs zwischen 1 und 20 Instances CORE oder

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
		Instance-Flotteneinheiten auf oder TASK Knoten, die Spot verwenden. Amazon EMR skaliert nicht mit dem ON_DEMAND Typ.

Szenario 5: On-Demand-Instances auf Core-Knoten und Spot Instances auf Aufgabenknoten skalieren

Um On-Demand-Instances auf Core-Knoten und Spot Instances auf Aufgabenknoten zu skalieren, müssen die verwalteten Skalierungsparameter die folgenden Anforderungen erfüllen:

- Das On-Demand-Limit muss dem maximalen Core-Knoten entsprechen.
- Sowohl das On-Demand-Limit als auch die maximale Anzahl an Core-Knoten müssen unter der maximalen Grenze liegen.

Um dieses Szenario in einem Cluster zu aktivieren, der aus Instance-Gruppen besteht, muss die Core-Knotengruppe die On-Demand-Kaufoption verwenden.

Dieses Szenario ist nicht anwendbar, wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden und Ihre Anwendungsprozesse darauf beschränken, nur auf ON_DEMAND Knoten oder CORE Knoten ausgeführt zu werden.

Die folgenden Beispiele veranschaulichen das Szenario der Skalierung von On-Demand-Instances auf Core-Knoten und Spot Instances auf Aufgabenknoten.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen		

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Core: 1 On-Demand Aufgabe: 1 On-Demand- und 1 Spot	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 7 MaximumCoreCapacityUnits : 7	Hochskalieren Sie auf bis zu 6 On-Demand -Einheiten auf dem Core-Knoten, da sich bereits 1 On-Demand-Einheit auf dem Aufgabenknoten befindet und die maximale Anzahl für On-Demand -Einheiten 7 beträgt.
Instance-Flotten Core: 1 On-Demand Aufgabe: 1 On-Demand- und 1 Spot	UnitType: InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 7 MaximumCoreCapacityUnits : 7	Hochskalieren Sie anschließend auf bis zu 13 Spot-Einheiten auf Aufgabenknoten.

Szenario 6: Skalieren Sie **CORE** Instances für den Bedarf von Anwendungsprozessen und **TASK** Instanzen für den Bedarf von Executoren.

Dieses Szenario ist nur anwendbar, wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden und Anwendungsprozesse so einschränken, dass sie nur auf CORE Knoten ausgeführt werden.

Um CORE Knoten auf der Grundlage der Anforderungen des Anwendungsprozesses und TASK Knoten auf der Grundlage der Anforderungen der Executoren zu skalieren, müssen Sie beim Clusterstart die folgenden Konfigurationen festlegen:

- `yarn.node-labels.enabled:true`

- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

Wenn Sie den ON_DEMAND Grenzwert und den maximalen CORE Knotenparameter nicht angeben, verwenden beide Parameter standardmäßig die maximale Grenze.

Wenn die maximale Anzahl an ON_DEMAND Knoten kleiner als die maximale Grenze ist, verwendet die verwaltete Skalierung den maximalen ON_DEMAND Knotenparameter, um die Kapazitätszuweisung zwischen SPOT Knoten ON_DEMAND und Knoten aufzuteilen. Wenn Sie den Parameter für den maximalen CORE Knoten auf weniger als oder gleich dem Parameter für die minimale Kapazität festlegen, bleiben die CORE Knoten bei der maximalen Kernkapazität statisch.

Die folgenden Beispiele veranschaulichen das Szenario der Skalierung von CORE-Instanzen auf der Grundlage der Anforderungen des Anwendungsprozesses und von TASK-Instanzen auf der Grundlage der Nachfrage von Executoren.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen Core: 1 On-Demand Aufgabe: 1 On-Demand	UnitType: Instances MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 10 MaximumCoreCapacityUnits : 20	Skaliert CORE Knoten zwischen 1 und 20 Knoten basierend auf der Nachfrage nach dem Anwendungsprozess des Clusters unter Verwendung des Markttyps On-Demand oder Spot-Markt. Skaliert TASK Knoten auf der Grundlage der Nachfrage der Executoren und
Instance-Flotten Core: 1 On-Demand Aufgabe: 1 On-Demand	UnitType: InstanceFleetUnits MinimumCapacityUnits : 1 MaximumCapacityUnits : 20 MaximumOnDemandCapacityUnits : 10 MaximumCoreCapacityUnits : 20	Skaliert TASK Knoten auf der Grundlage der Nachfrage der Executoren und

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
		der verbleibenden verfügbaren Kapazität, nachdem Amazon EMR Knoten zugewiesen hat. CORE Die Summe der angeforderten TASK Knoten CORE und der Knoten wird den Wert von 20 nicht überschreiten. <code>maximumCapacity</code> Die Summe der angeforderten On-Demand-Core-Knoten und der On-Demand-Aufgabenknoten wird den Wert <code>maximumOnDemandCapacity</code> von 10 nicht überschreiten. Zusätzlich können Kern- oder Taskknoten verwenden den Spotmarkttyp.

Szenario 7: Skalieren Sie **ON_DEMAND** Instances für den Bedarf an Anwendungsprozessen und **SPOT** Instances für den Bedarf an Executoren.

Dieses Szenario ist nur anwendbar, wenn Sie verwaltete Skalierung mit Knotenbezeichnungen verwenden und Anwendungsprozesse so einschränken, dass sie nur auf ON_DEMAND Knoten ausgeführt werden.

Um ON_DEMAND Knoten auf der Grundlage der Anforderungen des Anwendungsprozesses und SPOT Knoten auf der Grundlage der Anforderungen der Executoren zu skalieren, müssen Sie beim Clusterstart die folgenden Konfigurationen festlegen:

- `yarn.node-labels.enabled:true`
- `yarn.node-labels.am.default-node-label-expression: 'ON_DEMAND'`

Wenn Sie den ON_DEMAND Grenzwert und den maximalen CORE Knotenparameter nicht angeben, verwenden beide Parameter standardmäßig die maximale Grenze.

Wenn die maximale Anzahl an CORE Knoten kleiner als die maximale Grenze ist, verwendet die verwaltete Skalierung den maximalen CORE Knotenparameter, um die Kapazitätszuweisung zwischen TASK Knoten CORE und Knoten aufzuteilen. Wenn Sie den Parameter für den maximalen CORE Knoten auf weniger als oder gleich dem Parameter für die minimale Kapazität festlegen, bleiben die CORE Knoten bei der maximalen Kernkapazität statisch.

Die folgenden Beispiele veranschaulichen das Szenario der Skalierung von On-Demand-Instances auf der Grundlage der Anforderungen des Anwendungsprozesses und Spot-Instances auf der Grundlage der Nachfrage von Executoren.

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
Instance-Gruppen	UnitType: Instances	Skaliert
Core: 1 On-Demand	MinimumCapacityUnits : 1	ON_DEMAND
Aufgabe: 1 On-Demand	MaximumCapacityUnits : 20	Knoten zwischen 1 und 20
	MaximumOnDemandCapacityUnits : 20	Knoten auf der Grundlage der Anforderungen

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
	<code>MaximumCoreCapacityUnits</code> : 10	des Clusters für den Anwendung
Instance-Flotten	<code>UnitType: InstanceFleetUnits</code>	sprozess unter Verwendung
Core: 1 On-Demand	<code>MinimumCapacityUnits</code> : 1	des CORE TASK Knotentyps oder.
Aufgabe: 1 On-Demand	<code>MaximumCapacityUnits</code> : 20	Skaliert SPOT Knoten auf der
	<code>MaximumOnDemandCapacityUnits</code> : 20	Grundlage der Nachfrage der
	<code>MaximumCoreCapacityUnits</code> : 10	Executoren und der verbleibe
		nden verfügbar en Kapazität , nachdem
		Amazon EMR Knoten zugewiesen hat.
		ON_DEMAND
		Die Summe der angeforderten
		SPOT Knoten ON_DEMAND
		und der Knoten wird den Wert
		von 20 nicht überschreiten.
		<code>maximumCapacity</code> Die
		Summe der angeforderten
		On-Demand -Core-Kno

Ausgangszustand des Clusters	Skalierungsparameter	Skalierungs-Verhalten
		ten und Spot-Core-Knoten wird den Wert <code>maximumCoreCapacity</code> von 10 nicht überschreiten. Zusätzliche On-Demand-Nodes oder Spot-Nodes verwenden den TASK Knotentyp.

Grundlegendes zu verwalteten Skalierungsmetriken in Amazon EMR

Amazon EMR veröffentlicht hochaufgelöste Metriken mit Daten mit einer Granularität von einer Minute, wenn die verwaltete Skalierung für einen Cluster aktiviert ist. Sie können Ereignisse bei jeder Initiierung und Beendigung der Größenänderung anzeigen, die durch verwaltete Skalierung mit der Amazon EMR-Konsole oder der CloudWatch Amazon-Konsole gesteuert werden. CloudWatch Metriken sind entscheidend für den Betrieb der von Amazon EMR verwalteten Skalierung. Wir empfehlen Ihnen, die CloudWatch Metriken genau zu überwachen, um sicherzustellen, dass keine Daten fehlen. Weitere Informationen darüber, wie Sie CloudWatch Alarmer konfigurieren können, um fehlende Messwerte zu erkennen, finden Sie unter [CloudWatch Amazon-Alarmer verwenden](#). Weitere Informationen zur Verwendung von CloudWatch Ereignissen mit Amazon EMR finden Sie unter [CloudWatchEreignisse überwachen](#).

Die folgenden Metriken geben die aktuelle oder Zielkapazitäten eines Clusters an. Diese Metriken sind nur verfügbar, wenn die verwaltete Skalierung aktiviert ist. Bei Clustern, die aus Instance-Flotten bestehen, werden die Cluster-Kapazitätsmetriken in `Units` gemessen. Bei Clustern, die aus Instance-Gruppen bestehen, werden die Clusterkapazitätsmetriken in `Nodes` oder `vCPU` basierend auf dem Einheitenentyp gemessen, der in der Richtlinie für verwaltete Skalierung verwendet wird.

Metrik	Beschreibung

Metrik	Beschreibung
• TotalUnitsRequested • TotalNodesRequested • TotalVCPURequested	Die Zielgesamtanzahl von units/nodes/vCPUs in einem Cluster, wie sie durch verwaltete Skalierung bestimmt wird. Einheiten: Anzahl
• TotalUnitsRunning • TotalNodesRunning • TotalVCPURunning	Die aktuelle Gesamtzahl der in einem laufenden Cluster units/nodes/vCPUs verfügbaren. Wenn eine Clustergrößenänderung angefordert wird, wird diese Metrik aktualisiert, nachdem die neuen Instances hinzugefügt oder aus dem Cluster entfernt wurden. Einheiten: Anzahl
• CoreUnitsRequested • CoreNodesRequested • CoreVCPURequested	Die Zielanzahl von CORE units/nodes/vCPUs in einem Cluster, wie sie durch verwaltete Skalierung bestimmt wird. Einheiten: Anzahl
• CoreUnitsRunning • CoreNodesRunning • CoreVCPURunning	Die aktuelle Anzahl von CORE, die in einem Cluster units/nodes/vCPUs ausgeführt werden. Einheiten: Anzahl
• TaskUnitsRequested • TaskNodesRequested • TaskVCPURequested	Die Zielanzahl von TASKs units/nodes/vCPUs in einem Cluster, wie sie durch verwaltete Skalierung bestimmt wird. Einheiten: Anzahl

Metrik	Beschreibung
- TaskUnitsRunning - TaskNodesRunning - TaskVCPURunning	Die aktuelle Anzahl von TASK, die in einem Cluster units/nodes/vCPUs ausgeführt werden. Einheiten: Anzahl

Die folgenden Metriken geben den Verwendungsstatus von Clustern und Anwendungen an. Diese Metriken sind für alle Amazon-EMR-Features verfügbar, werden jedoch mit einer höheren Auflösung mit Daten in einer einminütigen Granularität veröffentlicht, wenn die verwaltete Skalierung für einen Cluster aktiviert ist. Sie können die folgenden Metriken mit den Clusterkapazitätsmetriken in der vorherigen Tabelle korrelieren, um die Entscheidungen bezüglich der verwalteten Skalierung zu verständlich zu machen.

Metrik	Beschreibung
AppsCompleted	Anzahl der an YARN übermittelten abgeschlossenen Anwendungen. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
AppsPending	Anzahl der an YARN übermittelten Anwendungen, die sich im ausstehenden Zustand befinden. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
AppsRunning	Anzahl der an YARN übermittelten Anwendungen, die ausgeführt werden. Anwendungsfall: Cluster-Fortschritt überwachen

Metrik	Beschreibung
	Einheiten: Anzahl
ContainerAllocated	Die Anzahl der Ressourcencontainer, die von der zugewiesenen ResourceManager. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
ContainerPending	Anzahl der Container in der Warteschlange, die noch nicht zugeordnet worden sind. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
ContainerPendingRatio	Das Verhältnis von ausstehenden Containern zu zugewiesenen Containern ($\text{ContainerPendingRatio} = \text{ContainerPending} / \text{ContainerAllocated}$). Wenn $\text{ContainerAllocated} = 0$, dann $\text{ContainerPendingRatio} = \text{ContainerPending}$. Der Wert von ContainerPendingRatio steht für eine Zahl, nicht für einen Prozentsatz. Dieser Wert ist zum Skalieren von Cluster-Ressourcen anhand des Zuordnungsverhaltens des Containers hilfreich. Einheiten: Anzahl
HDFSUtilization	Prozentsatz des gegenwärtig benutzten HDFS-Speichers. Anwendungsfall: Cluster-Leistung analysieren Einheiten: Prozent

Metrik	Beschreibung
IsIdle	Gibt an, dass ein Cluster keine Arbeiten mehr ausführt, aber unverändert aktiv ist und Kosten verursacht. Der Wert beträgt 1, wenn weder Tasks noch Aufträge ausgeführt werden, andernfalls beträgt der Wert 0. Dieser Wert wird in 5-Minuten-Intervallen geprüft. Wenn der Wert 1 beträgt, bedeutet dies, dass der Cluster zum Zeitpunkt der Prüfung ungenutzt war, aber nicht die gesamten fünf Minuten. Um Fehlalarme zu vermeiden, sollten Sie einen Alarm auslösen, wenn dieser Wert mehrere aufeinander folgende fünfminütige Prüfungen lang 1 beträgt. Sie können zum Beispiel einen Alarm auslösen, wenn dieser Wert 30 Minuten oder länger 1 beträgt. Anwendungsfall: Cluster-Leistung überwachen Einheiten: boolescher Wert
MemoryAvailableMB	Verfügbarer zuzuordnender Speicher. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl
MRActiveNodes	Die Anzahl der Knoten, auf denen derzeit MapReduce Aufgaben oder Jobs ausgeführt werden. Entspricht der YARN-Metrik <code>mapred.resourcemanager.NoOfActiveNodes</code>. Anwendungsfall: Cluster-Fortschritt überwachen Einheiten: Anzahl

Metrik	Beschreibung
YARNMemoryAvailablePercentage	Der Prozentsatz des verbleibenden Speichers, der YARN zur Verfügung steht ($\text{YARNMemoryAvailablePercentage} = \text{MemoryAvailable MB} / \text{MemoryTotalMB}$). Dieser Wert ist zum Skalieren von Cluster-Ressourcen anhand der YARN-Speichernutzung hilfreich. Einheiten: Prozent

Die folgenden Metriken enthalten Informationen zu den Ressourcen, die von YARN-Containern und -Knoten verwendet werden. Diese Metriken aus dem YARN-Ressourcenmanager bieten Einblicke in die Ressourcen, die von Containern und Knoten verwendet werden, die im Cluster ausgeführt werden. Der Vergleich dieser Metriken mit den Clusterkapazitätsmetriken der vorherigen Tabelle bietet ein klareres Bild der Auswirkungen der verwalteten Skalierung:

Metrik	Dazugehörige Veröffentlichungen	Beschreibung
YarnContainersUsedMemoryGBSeconds	Verfügbar für Release-Label 7.3.0 und höher	Der verbrauchte Container-Speicher in * Sekunden für den Veröffentlichungszeitraum. Einheiten: GB * Sekunden
YarnContainersTotalMemoryGBSeconds	Verfügbar für Release-Label 7.3.0 und höher	Der gesamte Garnbehälter (* Sekunden) für den Veröffentlichungszeitraum. Einheiten: GB * Sekunden
YarnContainersUsedVCPUSeconds	Verfügbar für Release-Label 7.5.0 und höher	Die verbrauchte Container-VCPU (* Sekunden) für den Veröffentlichungszeitraum.

Metrik	Dazugehörige Veröffentlichungen	Beschreibung
		Einheiten: VCPU * Sekunden
YarnContainersTotalVCPUSeconds	Verfügbar für Release-Label 7.5.0 und höher	Die Gesamtanzahl der Container-VCPU* Sekunden für den Veröffentlichungszeitraum. Einheiten: VCPU * Sekunden
YarnNodesUsedMemoryGBSeconds	Verfügbar für Release-Label 7.5.0 und höher	Der verbrauchte Knotenspeicher (* Sekunden) für den Veröffentlichungszeitraum. Einheiten: GB * Sekunden
YarnNodesTotalMemoryGBSeconds	Verfügbar für Release-Label 7.5.0 und höher	Der gesamte Knotenspeicher in * Sekunden für den Veröffentlichungszeitraum. Einheiten: GB * Sekunden
YarnNodesUsedVCPUSeconds	Verfügbar für Release-Label 7.3.0 und höher	Die verbrauchte Knoten-VCPU in * Sekunden für den Veröffentlichungszeitraum. Einheiten: VCPU * Sekunden
YarnNodesTotalVCPUSeconds	Verfügbar für Release-Label 7.3.0 und höher	Die Gesamtzahl der Knoten-VCPU* Sekunden für den Veröffentlichungszeitraum. Einheiten: VCPU * Sekunden

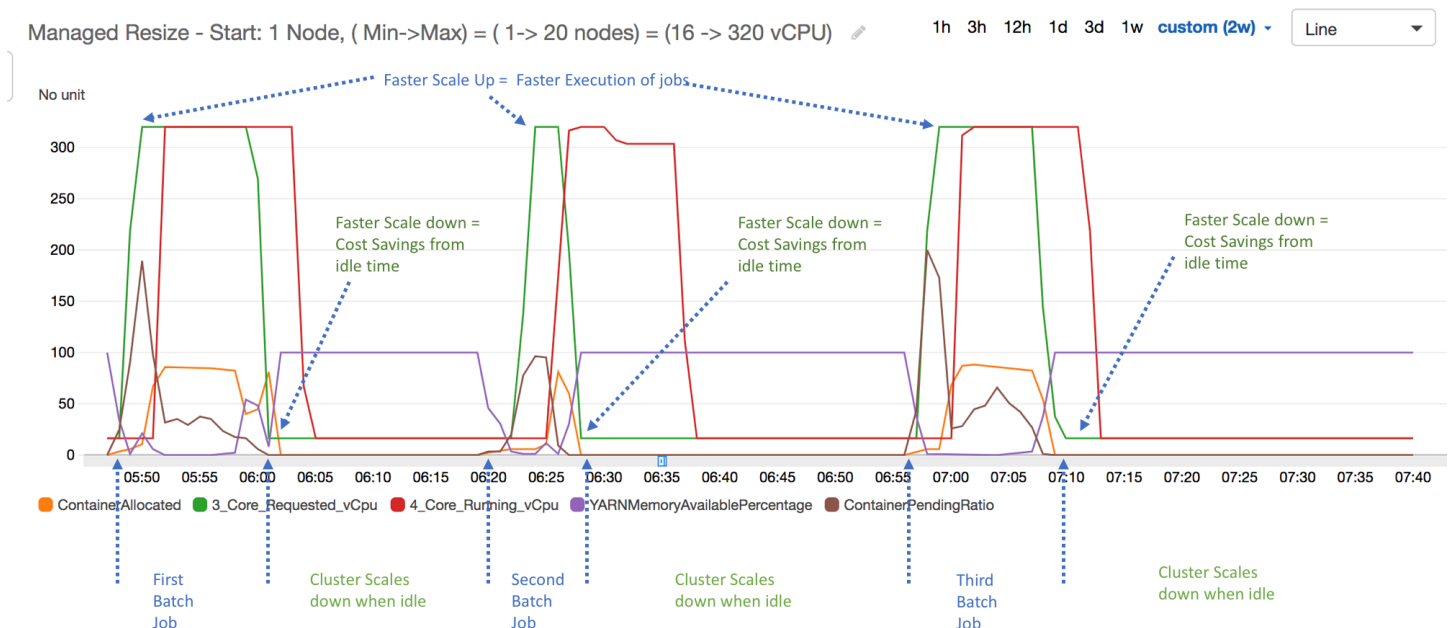
Grafieren der Metriken für verwaltete Skalierung

Sie können Metriken als Grafiken darstellen, um die Workload-Muster Ihres Clusters und entsprechenden Skalierungsentscheidungen zu visualisieren, die durch Amazon EMR Managed Scaling getroffen werden, wie die folgenden Schritte veranschaulichen.

Um verwaltete Skalierungsmetriken in der CloudWatch Konsole grafisch darzustellen

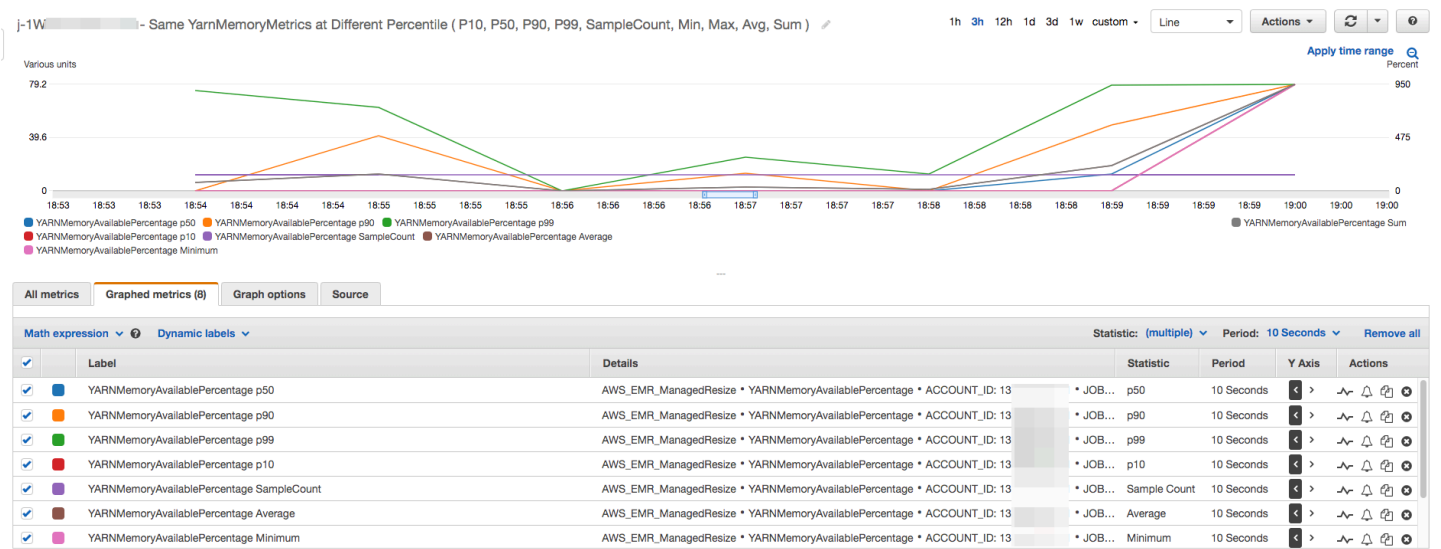
1. Öffnen Sie die [CloudWatch-Konsole](#).
2. Wählen Sie im Navigationsbereich Amazon EMR aus. Sie können die Cluster-Kennung auch nach dem zu überwachenden Cluster durchsuchen.
3. Scrollen Sie zur Metrik, die grafisch dargestellt werden soll. Öffnen Sie eine Metrik, um das Diagramm anzuzeigen.
4. Um eine oder mehrere Metriken grafisch darzustellen, aktivieren Sie das Kontrollkästchen neben jeder Metrik.

Das folgende Beispiel veranschaulicht die Aktivität von Amazon EMR Managed Scaling eines Clusters. Das Diagramm zeigt drei automatische Scale-Down-Perioden, die Kosten sparen, wenn eine weniger aktive Workload vorliegt.



Alle Cluster-Kapazitäts- und Nutzungsmetriken werden in Intervallen von einer Minute veröffentlicht. Zusätzliche statistische Informationen sind auch jeweils mit allen einminütigen Daten verknüpft, sodass Sie verschiedene Funktionen wie Percentiles, Min, Max, Sum, Average, SampleCount darstellen können.

Im folgenden Diagramm wird beispielsweise dieselbe YARNMemoryAvailablePercentage-Metrik an verschiedenen Perzentilen (P10, P50, P90, P99) zusammen mit Sum, Average, Min, SampleCount dargestellt.



Verwenden der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instanzgruppen in Amazon EMR

Die automatische Skalierung mit einer benutzerdefinierten Richtlinie in Amazon EMR-Versionen 4.0 und höher ermöglicht Ihnen die programmatische Skalierung und Skalierung in Kern- und Aufgabenknoten auf der Grundlage einer CloudWatch Metrik und anderer Parameter, die Sie in einer Skalierungsrichtlinie angeben. Automatische Skalierung mit einer benutzerdefinierten Richtlinie ist bei der Instance-Gruppenkonfiguration verfügbar, aber nicht bei der Verwendung von Instance-Flotten. Weitere Informationen zu Instance-Gruppen und Instance-Flotten finden Sie unter [Erstellen Sie einen Amazon EMR-Cluster mit Instance-Flotten oder einheitlichen Instance-Gruppen](#).

Note

Um das Auto Scaling mit einem benutzerdefinierten Richtlinienfeature in Amazon EMR zu verwenden, müssen Sie beim Erstellen eines Clusters `true` für den

Parameter `VisibleToAllUsers` festlegen. Weitere Informationen finden Sie unter [SetVisibleToAllUsers](#).

Die Skalierungsrichtlinie ist Teil einer Instance-Gruppen-Konfiguration. Sie können eine Richtlinie während der anfänglichen Konfiguration einer Instance-Gruppe oder durch Ändern einer Instance-Gruppe in einer vorhandenen Cluster-Gruppe festlegen (auch wenn die Instance aktiv ist). Jede Instance-Gruppe in einem Cluster, mit Ausnahme der Primär-Instance-Gruppe, kann ihre eigene Skalierungsrichtlinie haben. Diese besteht aus Regeln zur Hoch- und Herunter-Skalierung. Scale-Out- und Scale-In-Regeln können unabhängig konfiguriert werden. Jede Regel kann andere Parameter haben.

Sie können Skalierungsrichtlinien mit der AWS Management Console, der AWS CLI, der oder der Amazon EMR-API konfigurieren. Wenn Sie die AWS CLI oder die Amazon EMR-API verwenden, geben Sie die Skalierungsrichtlinie im JSON-Format an. Wenn Sie die AWS CLI oder die Amazon EMR-API verwenden, können Sie außerdem benutzerdefinierte CloudWatch Metriken angeben. Benutzerdefinierte Metriken können nicht über die AWS Management Console ausgewählt werden. Wenn Sie eine Skalierungsrichtlinie zum ersten Mal über die Konsole erstellen, wird eine für viele Anwendungen geeignete Standardrichtlinie erstellt. Diese können Sie als Basis für Ihre eigene Richtlinie nutzen. Sie können die Standardregeln löschen oder ändern.

Auch wenn die automatische Skalierung es Ihnen ermöglicht, die EMR-Clusterkapazität anzupassen *on-the-fly*, sollten Sie dennoch die grundlegenden Workload-Anforderungen berücksichtigen und Ihre Knoten- und Instanzgruppenkonfigurationen planen. Weitere Informationen finden Sie unter [Richtlinien zur Cluster-Konfiguration](#).

Note

Bei den meisten Workloads ist die Einrichtung von Scale-In- und Scale-Out-Regeln zur Optimierung der Ressourcenauslastung erstrebenswert. Wenn Sie eine Regel ohne Gegenstück erstellen, müssen Sie die Größe der Instanz nach einer Skalierung möglicherweise manuell anpassen. In diesem Fall richten Sie sozusagen ein "unidirektionales" Auto Scaling in eine Richtung (Scale-Out oder Scale-In) mit einem manuellen Reset ein.

Erstellen einer IAM-Rolle zur automatischen Skalierung

Auto Scaling in Amazon EMR erfordert eine IAM-Rolle mit Berechtigungen zum Hinzufügen und Beenden von Instances für den Fall, dass die Skalierung ausgelöst wird. Eine Standardrolle, `EMR_AutoScaling_DefaultRole`, mit der entsprechenden Rollen- und Vertrauensrichtlinie, ist für diesen Zweck verfügbar. Wenn Sie mit dem zum ersten Mal einen Cluster mit einer Skalierungsrichtlinie erstellen AWS Management Console, erstellt Amazon EMR die Standardrolle und fügt die verwaltete Standardrichtlinie für Berechtigungen an. `AmazonElasticMapReduceforAutoScalingRole`

Wenn Sie einen Cluster mit einer automatischen Skalierungsrichtlinie mit dem erstellen AWS CLI, müssen Sie zunächst sicherstellen, dass entweder die Standard-IAM-Rolle vorhanden ist oder dass Sie über eine benutzerdefinierte IAM-Rolle mit einer angehängten Richtlinie verfügen, die die entsprechenden Berechtigungen bereitstellt. Um die Standardrolle zu erstellen, können Sie den Befehl `create-default-roles` ausführen, bevor Sie einen Cluster erstellen. Sie können dann die Option `--auto-scaling-role EMR_AutoScaling_DefaultRole` angeben, wenn Sie einen Cluster erstellen. Alternativ können Sie eine benutzerdefinierte Rolle mit Auto Scaling erstellen und diese angeben, wenn Sie einen Cluster erstellen, zum Beispiel `--auto-scaling-role MyEMRAutoScalingRole`. Wenn Sie eine benutzerdefinierte Auto-Scaling-Rolle für Amazon EMR erstellen, empfehlen wir Ihnen, die Berechtigungsrichtlinien für Ihre benutzerdefinierte Rolle auf der Grundlage der verwalteten Richtlinie zu erstellen. Weitere Informationen finden Sie unter [Konfigurieren Sie IAM-Servicerollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen](#).

Grundlegendes zu Auto-Scaling-Regeln

Wenn eine Scale-Out-Regel eine Skalierungsaktivität für eine Instance-Gruppe auslöst, werden EC2 Amazon-Instances gemäß Ihren Regeln zur Instance-Gruppe hinzugefügt. Neue Knoten können von Anwendungen wie Apache Spark, Apache Hive und Presto verwendet werden, sobald die EC2 Amazon-Instance in den `InService` Status wechselt. Sie können außerdem eine Scale-In-Regel erstellen, die Instances beendet und Knoten entfernt. Weitere Informationen zum Lebenszyklus von EC2 Amazon-Instances, die automatisch skalieren, finden Sie unter [Auto Scaling Scaling-Lebenszyklus](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.

Sie können konfigurieren, wie ein Cluster EC2 Amazon-Instances beendet. Sie können wählen, ob Sie entweder innerhalb der EC2 Amazon-Instance-Stundengrenze für die Abrechnung oder nach Abschluss der Aufgabe kündigen möchten. Diese Einstellung gilt sowohl für die Auto Scaling- als

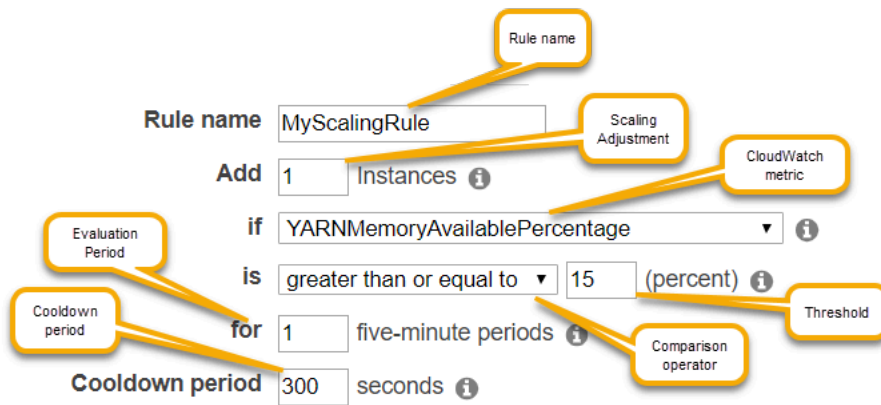
auch für manuelle Größenanpassungen. Weitere Informationen zu dieser Konfiguration finden Sie unter [Cluster-Scale-Down-Optionen für Amazon EMR-Cluster](#).

Die folgenden Parameter für eine Regel in einer Richtlinie bestimmen das Auto Scaling-Verhalten.

 Note

Die hier aufgeführten Parameter basieren auf dem AWS Management Console für Amazon EMR. Wenn Sie die AWS CLI oder Amazon EMR-API verwenden, sind zusätzliche erweiterte Konfigurationsoptionen verfügbar. Weitere Informationen zu erweiterten Optionen finden Sie [SimpleScalingPolicyConfiguration](#) in der Amazon EMR API-Referenz.

- Maximale und minimale Instances-Anzahl. Die Beschränkung Maximum Instances gibt die maximale Anzahl von EC2 Amazon-Instances an, die sich in der Instance-Gruppe befinden können, und gilt für alle Scale-Out-Regeln. In ähnlicher Weise gibt die Beschränkung „Minimum Instances“ die Mindestanzahl von EC2 Amazon-Instances an und gilt für alle Scale-In-Regeln.
- Der Rule name (Regelname) muss innerhalb der Richtlinie eindeutig sein.
- Die Skalierungsanpassung, die die Anzahl der EC2 Instances bestimmt, die während der durch die Regel ausgelösten Skalierungsaktivität hinzugefügt (für Scale-Out-Regeln) oder beendet (für Scale-In-Regeln) werden.
- Die CloudWatch Metrik, die im Hinblick auf einen Alarm überwacht wird.
- Ein Vergleichsoperator, der verwendet wird, um die CloudWatch Metrik mit dem Schwellenwert zu vergleichen und eine Auslösebedingung zu bestimmen.
- Ein Bewertungszeitraum in Schritten von fünf Minuten, für den sich die CloudWatch Metrik in einem Triggerzustand befinden muss, bevor die Skalierungsaktivität ausgelöst wird.
- Eine Ruhephase in Sekunden legt fest, wie viel Zeit zwischen einer durch eine Regel ausgelösten Skalierung und dem Start der nächsten Skalierung vergehen muss (unabhängig von der auslösenden Regel). Wenn eine Instanzgruppe eine Skalierungsaktivität abgeschlossen hat und ihren Status nach der Skalierung erreicht hat, bietet die Abklingzeit die Möglichkeit, dass sich die CloudWatch Metriken, die nachfolgende Skalierungsaktivitäten auslösen könnten, stabilisieren. Weitere Informationen finden Sie unter [Auto Scaling Scaling-Abklingzeiten](#) im Amazon EC2 Auto Scaling Scaling-Benutzerhandbuch.



Überlegungen und Einschränkungen

- CloudWatch Amazon-Metriken sind entscheidend für den Betrieb der automatischen Skalierung von Amazon EMR. Wir empfehlen Ihnen, die CloudWatch Amazon-Metriken genau zu beobachten, um sicherzustellen, dass keine Daten fehlen. Weitere Informationen darüber, wie Sie CloudWatch Amazon-Alarme konfigurieren können, um fehlende Messwerte zu erkennen, finden Sie unter [CloudWatch Amazon-Alarme verwenden](#).
- Eine übermäßige Auslastung von EBS-Volumes kann zu Problemen bei der verwalteten Skalierung führen. Wir empfehlen, die Nutzung des EBS-Volumes genau zu überwachen, um sicherzustellen, dass das EBS-Volume unter 90 % ausgelastet ist. Informationen zur Angabe zusätzlicher EBS-Volumes finden Sie unter [Instance-Speicher](#).
- Bei der automatischen Skalierung mit einer benutzerdefinierten Richtlinie in den Amazon EMR-Versionen 5.18 bis 5.28 kann es zu Skalierungsfehlern kommen, die durch zeitweise fehlende Daten in den Amazon-Metriken verursacht werden. CloudWatch Wir empfehlen, dass Sie die neuesten Amazon-EMR-Versionen verwenden, um das Auto Scaling zu verbessern. Sie können sich auch an den [AWS -Support](#) wenden, um einen Patch zu erhalten, wenn Sie eine Amazon-EMR-Version zwischen 5.18 und 5.28 verwenden müssen.

Verwenden von, um die automatische Skalierung zu konfigurieren AWS Management Console

Wenn Sie einen Cluster erstellen, konfigurieren Sie mithilfe der erweiterten Optionen für die Cluster-Konfiguration eine Skalierungsrichtlinie für Instance-Gruppen. Sie können außerdem eine Skalierungsrichtlinie für eine laufende Instance-Gruppe erstellen, indem Sie die Hardware-Einstellungen eines vorhandenen Clusters bearbeiten.

1. Navigieren Sie zur neuen Amazon-EMR-Konsole und wählen Sie in der Seitennavigation die Option Zur alten Konsole wechseln aus. Weitere Informationen darüber, was Sie erwartet, wenn Sie zur alten Konsole wechseln, finden Sie unter [Verwenden der alten Konsole](#).
2. Wenn Sie einen Cluster in der Amazon-EMR-Konsole erstellen, wählen Sie Cluster erstellen und Zu erweiterten Optionen gehen aus, wählen dann Optionen für Schritt 1: Software und Schritte aus und wechseln dann zu Schritt 2: Hardware-Konfiguration.

– oder –

Wenn Sie eine Instance-Gruppe in einem ausgeführten Cluster ändern, wählen Sie den Cluster aus der Cluster-Liste aus und erweitern Sie dann den Hardware-Abschnitt.

3. Wählen Sie im Abschnitt Clusterskalierung und Bereitstellungsoption die Option Clusterskalierung aktivieren aus. Wählen Sie dann Benutzerdefinierte Richtlinie für automatische Skalierung erstellen aus.

Klicken Sie in der Tabelle Benutzerdefinierte Richtlinien für automatische Skalierung auf das Stiftsymbol in der Zeile der Instance-Gruppe, die Sie konfigurieren möchten. Die Seite Auto-Scaling-Regeln wird geöffnet.

4. Geben Sie die Maximum instances (maximale Instances)-Anzahl ein, die die Instance-Gruppe nach dem Scale-Out enthalten soll. Geben Sie die Minimum instances (minimale Instances)-Anzahl ein, die die Instance-Gruppe nach dem Scale-In enthalten soll.
5. Klicken Sie auf den Stift, um Regelparameter zu bearbeiten. Klicken Sie auf das X, um eine Regel aus der Richtlinie zu entfernen, und klicken Sie auf Add rule (Regel hinzufügen), um weitere Regeln hinzuzufügen.
6. Wählen Sie die weiter oben in diesem Thema beschriebenen Regelparameter aus. Eine Beschreibung der verfügbaren CloudWatch Metriken für Amazon EMR finden Sie unter [Amazon EMR-Metriken und -Dimensionen](#) im CloudWatch Amazon-Benutzerhandbuch.

Verwenden von, AWS CLI um die automatische Skalierung zu konfigurieren

Sie können AWS CLI Befehle für Amazon EMR verwenden, um die automatische Skalierung zu konfigurieren, wenn Sie einen Cluster und eine Instanzgruppe erstellen. Sie können eine Kurzschreibweise mit der passenden JSON-Konfiguration in den entsprechenden Befehlen oder eine Referenzdatei mit der JSON-Konfiguration verwenden. Sie können außerdem eine Auto Scaling-Richtlinie auf eine vorhandene Instance-Gruppe anwenden und eine angewendete Auto Scaling-

Richtlinie entfernen. Darüber hinaus können Sie Details einer Skalierungsrichtlinien-Konfiguration aus einem aktuell ausgeführten Cluster abrufen.

Important

Wenn Sie einen Cluster erstellen, der eine Richtlinie für automatische Skalierung nutzt, müssen Sie mit dem Befehl `--auto-scaling-role` *MyAutoScalingRole* die IAM-Rolle für das Auto Scaling anzugeben. Die Standard-Rolle ist *EMR_AutoScaling_DefaultRole* und kann mit dem Befehl `create-default-roles` erstellt werden. Die Rolle kann nur hinzugefügt werden, wenn der Cluster erstellt wird. Sie kann nicht zu einem vorhandenen Cluster hinzugefügt werden.

Eine ausführliche Beschreibung der Parameter, die bei der Konfiguration einer automatischen Skalierungsrichtlinie verfügbar sind, finden Sie [PutAutoScalingPolicy](#) in Amazon EMR API Reference.

Erstellen eines Clusters mit einer angewendeten Auto-Scaling-Richtlinie in einer Instance-Gruppe

Sie können eine Auto Scaling-Konfiguration innerhalb der Option `--instance-groups` des Befehls `aws emr create-cluster` festlegen. Das folgende Beispiel demonstriert einen create-Cluster-Befehl, in dem eine Auto Scaling-Richtlinie für die Core-Instance-Gruppe enthalten ist. Der Befehl erstellt eine Skalierungskonfiguration, die der Standard-Scale-Out-Richtlinie entspricht, die angezeigt wird, wenn Sie eine automatische Skalierungsrichtlinie mit dem AWS Management Console für Amazon EMR erstellen. Aus Gründen der Übersichtlichkeit verzichten wir auf die Abbildung einer Scale-In-Richtlinie. Das Erstellen einer Scale-Out-Regel ohne Verringern der Scale-In-Regel wird nicht empfohlen.

```
aws emr create-cluster --release-label emr-5.2.0 --service-role
  EMR_DefaultRole --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole
  --auto-scaling-role EMR_AutoScaling_DefaultRole --instance-groups
  Name=MyMasterIG,InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1
  'Name=MyCoreIG,InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,AutoScalingPolicy
scale-out,Description=Replicates the default scale-out rule in the
console.,Action={SimpleScalingPolicyConfiguration={AdjustmentType=CHANGE_IN_CAPACITY,ScalingAd
ElasticMapReduce,Period=300,Statistic=AVERAGE,Threshold=15,Unit=PERCENT,Dimensions=[{Key=JobFlo
```

Der folgende Befehl veranschaulicht die Verwendung der Befehlszeile zur Angabe einer Auto-Scaling-Richtliniendefinition im Rahmen einer Instance-Gruppe-Konfigurationsdatei mit dem Namen *instancegroupconfig.json*.

```
aws emr create-cluster --release-label emr-5.2.0 --service-role EMR_DefaultRole --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole --instance-groups file://your/path/to/instancegroupconfig.json --auto-scaling-role EMR_AutoScaling_DefaultRole
```

Der Inhalt der Konfigurationsdatei sieht wie folgt aus:

```
[
{
  "InstanceCount": 1,
  "Name": "MyMasterIG",
  "InstanceGroupType": "MASTER",
  "InstanceType": "m5.xlarge"
},
{
  "InstanceCount": 2,
  "Name": "MyCoreIG",
  "InstanceGroupType": "CORE",
  "InstanceType": "m5.xlarge",
  "AutoScalingPolicy":
  {
    "Constraints":
    {
      "MinCapacity": 2,
      "MaxCapacity": 10
    },
    "Rules":
    [
      {
        "Name": "Default-scale-out",
        "Description": "Replicates the default scale-out rule in the console for YARN
memory.",
        "Action":{
          "SimpleScalingPolicyConfiguration":{
            "AdjustmentType": "CHANGE_IN_CAPACITY",
            "ScalingAdjustment": 1,
            "CoolDown": 300
          }
        },
        "Trigger":{
          "CloudWatchAlarmDefinition":{
            "ComparisonOperator": "LESS_THAN",
            "EvaluationPeriods": 1,
```

```

    "MetricName": "YARNMemoryAvailablePercentage",
    "Namespace": "AWS/ElasticMapReduce",
    "Period": 300,
    "Threshold": 15,
    "Statistic": "AVERAGE",
    "Unit": "PERCENT",
    "Dimensions": [
      {
        "Key": "JobFlowId",
        "Value": "${emr.clusterId}"
      }
    ]
  }
}
]

```

Hinzufügen einer Instance-Gruppe mit einer Auto-Scaling-Richtlinie zu einem Cluster

Sie können genauso wie bei `--instance-groups` mithilfe der `add-instance-groups`-Option des `create-cluster`-Befehls eine Skalierungsrichtlinien-Konfiguration festlegen. Im folgenden Beispiel wird ein Verweis auf eine JSON-Datei (*instancegroupconfig.json*) mit der Konfiguration der Instance-Gruppe verwendet.

```
aws emr add-instance-groups --cluster-id j-1EKZ3TYEVF1S2 --instance-groups file://your/path/to/instancegroupconfig.json
```

Anwenden einer Auto-Scaling-Richtlinie auf eine vorhandene Instance-Gruppe oder Ändern einer angewandten Richtlinie

Verwenden Sie den `aws emr put-auto-scaling-policy`-Befehl, um eine Auto Scaling-Richtlinie auf eine vorhandene Instance-Gruppe anzuwenden. Die Instance-Gruppe muss Teil eines Clusters sein, der die Auto-Scaling-IAM-Rolle verwendet. Im folgenden Beispiel wird ein Verweis auf eine JSON-Datei (*autoscaleconfig.json*) verwendet, in der eine Auto Scaling-Richtlinienkonfiguration definiert ist.

```
aws emr put-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-id ig-3PLUZBA6WLS07 --auto-scaling-policy file://your/path/to/autoscaleconfig.json
```

Der Inhalt der `autoscaleconfig.json`-Datei, die die gleiche Scale-Out-Regel wie im vorherigen Beispiel definiert, ist unten dargestellt.

```
{
  "Constraints": {
    "MaxCapacity": 10,
    "MinCapacity": 2
  },
  "Rules": [{
    "Action": {
      "SimpleScalingPolicyConfiguration": {
        "AdjustmentType": "CHANGE_IN_CAPACITY",
        "CoolDown": 300,
        "ScalingAdjustment": 1
      }
    },
    "Description": "Replicates the default scale-out rule in the console
for YARN memory",
    "Name": "Default-scale-out",
    "Trigger": {
      "CloudWatchAlarmDefinition": {
        "ComparisonOperator": "LESS_THAN",
        "Dimensions": [{
          "Key": "JobFlowId",
          "Value": "${emr.clusterID}"
        }],
        "EvaluationPeriods": 1,
        "MetricName": "YARNMemoryAvailablePercentage",
        "Namespace": "AWS/ElasticMapReduce",
        "Period": 300,
        "Statistic": "AVERAGE",
        "Threshold": 15,
        "Unit": "PERCENT"
      }
    }
  ]
}
```

Entfernen einer Auto-Scaling-Richtlinie aus einer Instance-Gruppe

```
aws emr remove-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07
```


Abrufen einer Auto-Scaling-Richtlinienkonfiguration

Der `describe-cluster` Befehl ruft die Richtlinienkonfiguration im Block ab. InstanceGroup Mit dem folgenden Befehl wird beispielsweise die Konfiguration für den Cluster mit der Cluster-ID `j-1CW0HP4PI30VJ` abgerufen.

```
aws emr describe-cluster --cluster-id j-1CW0HP4PI30VJ
```

Der Befehl generiert die folgende Beispielausgabe:

```
{
  "Cluster": {
    "Configurations": [],
    "Id": "j-1CW0HP4PI30VJ",
    "NormalizedInstanceHours": 48,
    "Name": "Auto Scaling Cluster",
    "ReleaseLabel": "emr-5.2.0",
    "ServiceRole": "EMR_DefaultRole",
    "AutoTerminate": false,
    "TerminationProtected": true,
    "MasterPublicDnsName": "ec2-54-167-31-38.compute-1.amazonaws.com",
    "LogUri": "s3n://aws-logs-232939870606-us-east-1/elasticmapreduce/",
    "Ec2InstanceAttributes": {
      "Ec2KeyName": "performance",
      "AdditionalMasterSecurityGroups": [],
      "AdditionalSlaveSecurityGroups": [],
      "EmrManagedSlaveSecurityGroup": "sg-09fc9362",
      "Ec2AvailabilityZone": "us-east-1d",
      "EmrManagedMasterSecurityGroup": "sg-0bfc9360",
      "IamInstanceProfile": "EMR_EC2_DefaultRole"
    },
    "Applications": [
      {
        "Name": "Hadoop",
        "Version": "2.7.3"
      }
    ],
    "InstanceGroups": [
      {
        "AutoScalingPolicy": {
```

```

    "Status": {
      "State": "ATTACHED",
      "StateChangeReason": {
        "Message": ""
      }
    },
    "Constraints": {
      "MaxCapacity": 10,
      "MinCapacity": 2
    },
    "Rules": [
      {
        "Name": "Default-scale-out",
        "Trigger": {
          "CloudWatchAlarmDefinition": {
            "MetricName": "YARNMemoryAvailablePercentage",
            "Unit": "PERCENT",
            "Namespace": "AWS/ElasticMapReduce",
            "Threshold": 15,
            "Dimensions": [
              {
                "Key": "JobFlowId",
                "Value": "j-1CW0HP4PI30VJ"
              }
            ],
            "EvaluationPeriods": 1,
            "Period": 300,
            "ComparisonOperator": "LESS_THAN",
            "Statistic": "AVERAGE"
          }
        },
        "Description": "",
        "Action": {
          "SimpleScalingPolicyConfiguration": {
            "CoolDown": 300,
            "AdjustmentType": "CHANGE_IN_CAPACITY",
            "ScalingAdjustment": 1
          }
        }
      },
      {
        "Name": "Default-scale-in",
        "Trigger": {
          "CloudWatchAlarmDefinition": {

```

```

        "MetricName": "YARNMemoryAvailablePercentage",
        "Unit": "PERCENT",
        "Namespace": "AWS/ElasticMapReduce",
        "Threshold": 75,
        "Dimensions": [
            {
                "Key": "JobFlowId",
                "Value": "j-1CW0HP4PI30VJ"
            }
        ],
        "EvaluationPeriods": 1,
        "Period": 300,
        "ComparisonOperator": "GREATER_THAN",
        "Statistic": "AVERAGE"
    }
},
"Description": "",
"Action": {
    "SimpleScalingPolicyConfiguration": {
        "CoolDown": 300,
        "AdjustmentType": "CHANGE_IN_CAPACITY",
        "ScalingAdjustment": -1
    }
}
}
]
},
"Configurations": [],
"InstanceType": "m5.xlarge",
"Market": "ON_DEMAND",
"Name": "Core - 2",
"ShrinkPolicy": {},
"Status": {
    "Timeline": {
        "CreationDateTime": 1479413437.342,
        "ReadyDateTime": 1479413864.615
    },
    "State": "RUNNING",
    "StateChangeReason": {
        "Message": ""
    }
},
"RunningInstanceCount": 2,
"Id": "ig-3M16XBE8C3PH1",

```

```

        "InstanceGroupType": "CORE",
        "RequestedInstanceCount": 2,
        "EbsBlockDevices": []
    },
    {
        "Configurations": [],
        "Id": "ig-OP62I28NSE8M",
        "InstanceGroupType": "MASTER",
        "InstanceType": "m5.xlarge",
        "Market": "ON_DEMAND",
        "Name": "Master - 1",
        "ShrinkPolicy": {},
        "EbsBlockDevices": [],
        "RequestedInstanceCount": 1,
        "Status": {
            "Timeline": {
                "CreationDateTime": 1479413437.342,
                "ReadyDateTime": 1479413752.088
            },
            "State": "RUNNING",
            "StateChangeReason": {
                "Message": ""
            }
        },
        "RunningInstanceCount": 1
    }
],
"AutoScalingRole": "EMR_AutoScaling_DefaultRole",
"Tags": [],
"BootstrapActions": [],
"Status": {
    "Timeline": {
        "CreationDateTime": 1479413437.339,
        "ReadyDateTime": 1479413863.666
    },
    "State": "WAITING",
    "StateChangeReason": {
        "Message": "Cluster ready after last step completed."
    }
}
}
}

```

Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters

Sie können Instances zu Kern- und Task-Instance-Gruppen und Instance-Flotten in einem laufenden Cluster mit der AWS Management Console, AWS CLI, oder der Amazon EMR-API hinzufügen und daraus entfernen. Wenn ein Cluster Instance-Gruppen verwendet, müssen Sie die Anzahl der Instances explizit ändern. Wenn Ihr Cluster Instance-Flotten verwendet, können Sie die Zieleinheiten für On-Demand-Instances und Spot-Instances ändern. Die Instance-Flotte fügt anschließend Instances hinzu bzw. entfernt diese, um dem neuen Ziel zu entsprechen. Weitere Informationen finden Sie unter [Instance-Flotten-Optionen](#). Anwendungen können neu bereitgestellte EC2 Amazon-Instances zum Hosten von Knoten verwenden, sobald die Instances verfügbar sind. Wenn Instances entfernt werden, fährt Amazon EMR Aufgaben so herunter, dass es weder zu einer Unterbrechung der Aufträge noch zu einem Datenverlust kommt. Weitere Informationen finden Sie unter [Beendigung bei Aufgaben-Abschluss](#).

Die Größe eines Clusters mit der Konsole anpassen

Sie können die Größe eines Clusters über die Amazon-EMR-Konsole ändern.

Console

So ändern Sie die Anzahl der Instances für einen vorhandenen Cluster mithilfe der neuen Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR, dann die Option Clusters und dann den Cluster aus, den Sie aktualisieren möchten. Der Cluster muss ausgeführt werden. Sie können die Größe eines bereitgestellten oder beendeten Clusters nicht ändern.
3. Sehen Sie sich auf der Cluster-Detailseite auf der Registerkarte Instances den Bereich Instance-Gruppen an.
4. Um die Größe einer vorhandenen Instance-Gruppe zu ändern, wählen Sie das Optionsfeld neben der Core- oder Aufgaben-Instance-Gruppe aus, deren Größe Sie ändern möchten, und wählen Sie dann Größe der Instance-Gruppe ändern. Geben Sie die neue Anzahl der Instances für die Instance-Gruppe an und wählen Sie anschließend Größe ändern aus.

Note

Wenn Sie sich dafür entscheiden, die Größe einer laufenden Instance-Gruppe zu reduzieren, wählt Amazon EMR intelligent die Instances aus, die aus der Gruppe

entfernt werden sollen, um den Datenverlust zu minimieren. Für eine genauere Steuerung Ihrer Größenänderungsaktion können Sie die ID für die Instance-Gruppe auswählen, die Instances auswählen, die Sie entfernen möchten, und dann die Option Terminate verwenden. Weitere Informationen zum intelligenten Herunterskalierungs-Verhalten finden Sie unter [Cluster-Scale-Down-Optionen für Amazon EMR-Cluster](#).

5. Wenn Sie die Größenänderung abbrechen möchten, können Sie das Optionsfeld für eine Instance-Gruppe mit dem Status Größenänderung beenden auswählen und dann in der Liste der Aktionen die Option Größenänderung beenden auswählen.
6. Um Ihrem Cluster als Reaktion auf den steigenden Workload eine oder mehrere Aufgaben-Instance-Gruppen hinzuzufügen, wählen Sie Aufgaben-Instance-Gruppe hinzufügen aus der Liste der Aktionen aus. Wählen Sie den EC2 Amazon-Instance-Typ, geben Sie die Anzahl der Instances für die Aufgabengruppe ein und wählen Sie dann Task-Instance-Gruppe hinzufügen, um zum Bereich Instance-Gruppen für Ihren Cluster zurückzukehren.

Wenn Sie die Anzahl der Knoten ändern, wird der Status der Instance-Gruppe aktualisiert. Wenn die gewünschte Änderung abgeschlossen ist, ändert sich der Status zu Running (Wird ausgeführt).

Ändern Sie die Größe eines Clusters mit AWS CLI

Sie können den verwenden AWS CLI , um die Größe eines laufenden Clusters zu ändern. Sie können die Anzahl der Aufgabenknoten erhöhen oder verringern. Sie können außerdem die Anzahl der Core-Knoten in einem ausgeführten Cluster erhöhen oder verringern. Es ist auch möglich, eine Instanz in der Core-Instanzgruppe mit der AWS CLI oder der API herunterzufahren. Dies sollte mit Vorsicht erfolgen. Beim Beenden einer Instance in der Core-Instance-Gruppe besteht das Risiko eines Datenverlustes. Die Instance wird zudem nicht automatisch ersetzt.

Zusätzlich zur Größenanpassung der Core- und Aufgaben-Gruppen können Sie mithilfe der AWS CLI auch eine oder mehrere Aufgaben-Instance-Gruppen zu einem ausgeführten Cluster hinzufügen.

Um die Größe eines Clusters zu ändern, indem Sie die Anzahl der Instanzen mit dem AWS CLI

Sie können der Kerngruppe oder Aufgabengruppe Instanzen hinzufügen und mit dem AWS CLI `modify-instance-groups` Unterbefehl mit dem Parameter `Instances` aus der Aufgabengruppe entfernen. Erhöhen Sie `InstanceCount`, um Instances zu Core- oder Task-Gruppen hinzuzufügen. Reduzieren Sie `InstanceCount`, um die Anzahl der Instances in der

Gruppe zu verringern. Die Reduzierung der Anzahl der Instances einer Task-Gruppe auf null entfernt zwar alle Instances, nicht jedoch die Instance-Gruppe.

- Um die Anzahl der Instanzen in der Aufgaben-Instanzgruppe von 3 auf 4 zu erhöhen, geben Sie den folgenden Befehl ein und *ig-31JXXXXXXBT0* ersetzen Sie ihn durch die Instanzgruppen-ID.

```
aws emr modify-instance-groups --instance-groups  
InstanceGroupId=ig-31JXXXXXXBT0,InstanceCount=4
```

Verwenden Sie den Unterbefehl `InstanceGroupId`, um die `describe-cluster` abzurufen. Die Ausgabe ist ein JSON-Objekt mit dem Namen `Cluster`, das die ID jeder Instance-Gruppe enthält. Um diesen Befehl verwenden zu können, benötigen Sie die Cluster-ID (diese können Sie über den `aws emr list-clusters`-Befehl oder die Konsole abrufen). Um die Instanzgruppen-ID abzurufen, geben Sie den folgenden Befehl ein und *j-2AXXXXXXGAPLF* ersetzen Sie ihn durch die Cluster-ID.

```
aws emr describe-cluster --cluster-id j-2AXXXXXXGAPLF
```

Mit dem AWS CLI können Sie auch eine Instance in der Core-Instanzgruppe mit dem `--modify-instance-groups` Unterbefehl beenden.

Warning

Die Angabe von `EC2InstanceIdsToTerminate` muss mit Vorsicht erfolgen. Instances werden sofort beendet, unabhängig vom Status der Anwendungen, die auf ihnen ausgeführt werden, und die Instance wird nicht automatisch ersetzt. Dies gilt unabhängig von der Konfiguration vom Scale down behavior (Abwärtsskalierungsverhalten) für den Cluster. Wenn eine Instance auf diese Weise beendet wird, besteht das Risiko von Datenverlusten und unvorhersehbarem Clusterverhalten.

Um eine bestimmte Instanz zu beenden, benötigen Sie die Instanzgruppen-ID (vom `aws emr describe-cluster --cluster-id` Unterbefehl zurückgegeben) und die Instanz-ID (vom `aws emr list-instances --cluster-id` Unterbefehl zurückgegeben). Geben Sie den folgenden Befehl ein, ersetzen Sie ihn durch die Instanzgruppen-ID und *ig-6RXXXXXX07SA i-f9XXXXf2* ersetzen Sie ihn durch die Instanz-ID.

```
aws emr modify-instance-groups --instance-groups
  InstanceGroupId=ig-6RXXXXXX07SA,EC2InstanceIdsToTerminate=i-f9XXXXf2
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Um die Größe eines Clusters zu ändern, indem Sie Task-Instance-Gruppen hinzufügen, verwenden Sie AWS CLI

Mit dem AWS CLI können Sie einem Cluster 1—48 Task-Instance-Gruppen mit dem `--add-instance-groups` Unterbefehl hinzufügen. Aufgaben-Instances-Gruppen können nur zu einem Cluster mit einer Primär-Instance-Gruppe und einer Core-Instance-Gruppe hinzugefügt werden. Wenn Sie den verwenden AWS CLI, können Sie bei jeder Verwendung des Unterbefehls bis zu fünf Task-Instanzgruppen hinzufügen. `--add-instance-groups`

1. Um einem Cluster eine einzelne Task-Instanzgruppe hinzuzufügen, geben Sie den folgenden Befehl ein und *j-JXBXXXXXX37R* ersetzen Sie ihn durch die Cluster-ID.

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-groups
  InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
```

2. Um einem Cluster mehrere Task-Instance-Gruppen hinzuzufügen, geben Sie den folgenden Befehl ein und *j-JXBXXXXXX37R* ersetzen Sie ihn durch die Cluster-ID. Sie können bis zu fünf Task-Instance-Gruppen pro Befehl hinzufügen.

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-
groups InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
  InstanceCount=10,InstanceGroupType=task,InstanceType=m5.xlarge
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Unterbrechen einer Größenänderung

Wenn Sie Amazon-EMR-Version 4.1.0 oder höher verwenden, können Sie eine Größenänderung auch während einer laufenden Größenänderung durchführen. Sie können außerdem eine zuvor gesendete Anfrage zur Größenanpassung stoppen oder eine neue Anfrage senden, um eine frühere

Anfrage zu überschreiben, ohne auf deren Abschluss zu warten. Sie können eine vorhandene Größenänderung über die Konsole oder über einen `ModifyInstanceGroups`-API-Aufruf mit der aktuellen Anzahl als Zielanzahl für den Cluster beenden.

Die folgende Abbildung zeigt eine Task-Instance-Gruppe, deren Größe gerade geändert wird und bei der die Größenänderung über Stop (Stopp) beendet werden kann.



Um eine Größenänderung zu unterbrechen, verwenden Sie den AWS CLI

Sie können den verwenden AWS CLI , um eine Größenänderung mit dem Unterbefehl zu beenden. `modify-instance-groups` Nehmen wir an, Sie haben sechs Instances in der Instance-Gruppe und Sie möchten diese auf 10 erhöhen. Später entscheiden Sie, dass Sie diese Anforderung stornieren möchten:

- Die ursprüngliche Anforderung:

```
aws emr modify-instance-groups --instance-groups
  InstanceGroupId=ig-myInstanceGroupId, InstanceCount=10
```

Die zweite Anforderung zum Beenden der ersten Anforderung:

```
aws emr modify-instance-groups --instance-groups
  InstanceGroupId=ig-myInstanceGroupId, InstanceCount=6
```

Note

Da es sich um einen asynchronen Prozess handelt, ändert sich die Instance-Anzahl möglicherweise entsprechend der vorherigen API-Anforderung, bevor nachfolgende Anforderungen berücksichtigt werden. Bei einer Verkleinerung kann es sein, dass auf den Knoten noch Aufgaben ausgeführt werden. In diesem Fall wird die Instance-Gruppe nicht verkleinert, bis die Knoten ihre Arbeit abgeschlossen haben.

Suspendierter Zustand

Eine Instance-Gruppe geht in einen suspendierten Zustand über, wenn sie beim Versuch, die neuen Clusterknoten zu starten, auf zu viele Fehler stößt. Wenn z. B. neue Knoten während der

Durchführung von Bootstrap-Aktionen wiederholt fehlschlagen, wechselt die Instance-Gruppe in den Status `SUSPENDED`, statt weiterhin zu versuchen, neue Knoten bereitzustellen. Nachdem Sie das entsprechende Problem behoben haben, setzen Sie die Anzahl der gewünschten Knoten in der Instance-Gruppe des Clusters zurück. Anschließend fährt die Instance-Gruppe mit der Reservierung von Knoten fort. Das Ändern einer Instance-Gruppe weist Amazon EMR an, die Knotenbereitstellung erneut zu versuchen. Nicht ausgeführte Knoten werden neu gestartet oder beendet.

In dem AWS CLI gibt der `list-instances` Unterbefehl alle Instanzen und deren Status zurück, ebenso wie der Unterbefehl `describe-cluster`. Wenn Amazon EMR einen Fehler bei einer Instance-Gruppe erkennt, wird der Status der Gruppe auf `SUSPENDED` geändert.

Um einen Cluster im `SUSPENDED`-Zustand zurückzusetzen, verwenden Sie den AWS CLI

Geben Sie den `describe-cluster`-Unterbefehl mit dem Parameter `--cluster-id` ein, um den Status der Instances in Ihrem Cluster anzuzeigen.

- Um Informationen zu allen Instances und Instanzgruppen in einem Cluster anzuzeigen, geben Sie den folgenden Befehl ein und `j-3KVXXXXXXY7UG` ersetzen Sie ihn durch die Cluster-ID.

```
aws emr describe-cluster --cluster-id j-3KVXXXXXXY7UG
```

Die Ausgabe zeigt Informationen über Ihre Instance-Gruppen und den Status der Instances an:

```
{
  "Cluster": {
    "Status": {
      "Timeline": {
        "ReadyDateTime": 1413187781.245,
        "CreationDateTime": 1413187405.356
      },
      "State": "WAITING",
      "StateChangeReason": {
        "Message": "Waiting after step completed"
      }
    },
    "Ec2InstanceAttributes": {
      "Ec2AvailabilityZone": "us-west-2b"
    },
    "Name": "Development Cluster",
    "Tags": [],
    "TerminationProtected": false,
```

```

    "RunningAmiVersion": "3.2.1",
    "NormalizedInstanceHours": 16,
    "InstanceGroups": [
      {
        "RequestedInstanceCount": 1,
        "Status": {
          "Timeline": {
            "ReadyDateTime": 1413187775.749,
            "CreationDateTime": 1413187405.357
          },
          "State": "RUNNING",
          "StateChangeReason": {
            "Message": ""
          }
        },
        "Name": "MASTER",
        "InstanceGroupType": "MASTER",
        "InstanceType": "m5.xlarge",
        "Id": "ig-3ETXXXXXXFYV8",
        "Market": "ON_DEMAND",
        "RunningInstanceCount": 1
      },
      {
        "RequestedInstanceCount": 1,
        "Status": {
          "Timeline": {
            "ReadyDateTime": 1413187781.301,
            "CreationDateTime": 1413187405.357
          },
          "State": "RUNNING",
          "StateChangeReason": {
            "Message": ""
          }
        },
        "Name": "CORE",
        "InstanceGroupType": "CORE",
        "InstanceType": "m5.xlarge",
        "Id": "ig-3SUXXXXXXXQ9ZM",
        "Market": "ON_DEMAND",
        "RunningInstanceCount": 1
      }
    ],
    ...
  }

```

Um Informationen zu einer bestimmten Instance-Gruppe anzuzeigen, geben Sie den Unterbefehl `list-instances` mit den Parametern `--cluster-id` und `--instance-group-types` ein. Sie können Informationen für Primär-, Core- oder Aufgaben-Gruppen anzeigen.

```
aws emr list-instances --cluster-id j-3KVXXXXXXY7UG --instance-group-types "CORE"
```

Verwenden Sie den Unterbefehl `modify-instance-groups` mit dem Parameter `--instance-groups`, um einen Cluster mit dem `SUSPENDED`-Status zurückzusetzen. Die Instance-Gruppen-ID wird vom Unterbefehl `describe-cluster` zurückgegeben.

```
aws emr modify-instance-groups --instance-groups  
  InstanceGroupId=ig-3SUXXXXXXQ9ZM, InstanceCount=3
```

Überlegungen zur Reduzierung der Clustergröße

Wenn Sie sich dafür entscheiden, die Größe eines laufenden Clusters zu reduzieren, sollten Sie das folgende Verhalten und die folgenden Best Practices von Amazon EMR berücksichtigen:

- Um die Auswirkungen auf laufende Jobs zu reduzieren, wählt Amazon EMR intelligent die zu entfernenden Instances aus. Weitere Informationen zum Verhalten beim Herunterskalieren von Clustern finden Sie unter [Beendigung bei Aufgaben-Abschluss](#) im Verwaltungshandbuch für Amazon EMR.
- Wenn Sie die Größe eines Clusters herunterskalieren, kopiert Amazon EMR die Daten aus den Instances, die es entfernt hat, in die verbleibenden Instances. Stellen Sie sicher, dass in den Instances, die in der Gruppe verbleiben, ausreichend Speicherkapazität für diese Daten vorhanden ist.
- Amazon EMR versucht, HDFS auf Instances in der Gruppe außer Betrieb zu nehmen. Bevor Sie die Größe eines Clusters reduzieren, empfehlen wir, die HDFS-Schreib-I/O zu minimieren.
- Wenn Sie die Größe eines Clusters am genauesten steuern möchten, können Sie den Cluster in der Konsole anzeigen und zur Registerkarte Instances wechseln. Wählen Sie die ID für die Instance-Gruppe aus, deren Größe Sie ändern möchten. Verwenden Sie dann die Option `Terminate` für die spezifischen Instanceen, die Sie entfernen möchten.

Konfiguration von Bereitstellungs-Timeouts zur Steuerung der Kapazität in Amazon EMR

Wenn Sie Instanceflotten verwenden, können Sie Timeouts für die Bereitstellung konfigurieren. Ein Bereitstellungs-Timeout weist Amazon EMR an, die Bereitstellung von Instance-Kapazität zu beenden, wenn der Cluster während des Cluster-Starts oder der Cluster-Skalierung einen bestimmten Zeitschwellenwert überschreitet. In den folgenden Themen wird beschrieben, wie ein Bereitstellungs-Timeout für den Clusterstart und für Cluster-Hochskalierungsvorgänge konfiguriert wird.

Themen

- [Konfigurieren Sie Bereitstellungs-Timeouts für den Cluster-Start in Amazon EMR](#)
- [Ein Bereitstellungs-Timeout für den Cluster-Start in Amazon EMR anpassen](#)

Konfigurieren Sie Bereitstellungs-Timeouts für den Cluster-Start in Amazon EMR

Sie können einen Timeout-Zeitraum für die Bereitstellung von Spot Instances für jede Flotte in Ihrem Cluster definieren. Wenn Amazon EMR keine Spot-Kapazität bereitstellen kann, können Sie entweder den Cluster beenden oder stattdessen On-Demand-Kapazität bereitstellen. Wenn der Timeout-Zeitraum während der Cluster-Größenänderung endet, storniert Amazon EMR nicht bereitgestellte Spot-Anfragen. Nicht bereitgestellte Spot Instances werden nicht in On-Demand-Kapazität übertragen.

Führen Sie die folgenden Schritte aus, um ein Bereitstellungs-Timeout für den Clusterstart mit der Amazon-EMR-Konsole anzupassen.

Console

Um das Bereitstellungs-Timeout zu konfigurieren, wenn Sie einen Cluster mit der Konsole erstellen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Navigieren Sie auf der Seite Cluster erstellen zur Cluster-Konfiguration und wählen Sie Instanceflotten.

4. Geben Sie unter Option Clusterskalierung und -Bereitstellung die Spotgröße für Ihre Core- und Taskflotten an.
5. Wählen Sie unter Spot-Timeout-Konfiguration entweder Cluster nach Spot-Timeout beenden oder Nach Spot-Timeout zu On-Demand wechseln. Geben Sie dann den Timeout-Zeitraum für die Bereitstellung von Spot Instances an. Der Standardwert lautet 1 Stunde.
6. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
7. Um Ihren Cluster mit dem konfigurierten Timeout zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um ein Bereitstellungs-Timeout mit dem Befehl **create-cluster** anzugeben

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-XXXXX"]}' \
--instance-fleets
' [{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSpecification":{"OnDemandSpecification":{"AllocationStrategy":"lowest-price"}}, "InstanceTypeConfigs": [{"WeightedCapacity":1, "EbsConfiguration": {"EbsBlockDeviceConfigs": [{"VolumeSpecification": {"SizeInGB":32, "VolumeType":"gp2"}, "VolumesPerInstance":2}]}], "BidPriceAsPercentageOfOnDemandPrice": 1}],
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecification":{"SpotSpecification": {"TimeoutDurationMinutes":120, "TimeoutAction":"SWITCH_TO_ON_DEMAND"}, "OnDemandSpecification":{"AllocationStrategy":"lowest-price"}}, "InstanceTypeConfigs": [{"WeightedCapacity":1, "EbsConfiguration": {"EbsBlockDeviceConfigs": [{"VolumeSpecification": {"SizeInGB":32, "VolumeType":"gp2"}, "VolumesPerInstance":2}]}], "BidPriceAsPercentageOfOnDemandPrice": 2}] ]'
```

Ein Bereitstellungs-Timeout für den Cluster-Start in Amazon EMR anpassen

Definieren Sie einen Timeout-Zeitraum für die Bereitstellung von Spot Instances für jede Flotte in Ihrem Cluster. Wenn Amazon EMR die Spot-Kapazität nicht bereitstellen kann, storniert es die Größenänderungsanforderung und beendet seine Versuche, zusätzliche Spot-Kapazität

bereitzustellen. Wenn Sie einen Cluster erstellen, können Sie das Timeout konfigurieren. Für einen laufenden Cluster können Sie ein Timeout hinzufügen oder aktualisieren.

Wenn der Timeout-Zeitraum abläuft, sendet Amazon EMR Ereignisse automatisch an einen Amazon CloudWatch Events-Stream. Mit können Sie Regeln erstellen CloudWatch, die Ereignisse nach einem bestimmten Muster zuordnen, und die Ereignisse dann an Ziele weiterleiten, um Maßnahmen zu ergreifen. Sie können beispielsweise eine Regel zum Senden einer E-Mail-Benachrichtigung konfigurieren. Weitere Informationen zum Erstellen von Regeln finden Sie unter [Regeln für Amazon EMR-Ereignisse erstellen mit CloudWatch](#). Weitere Informationen zu verschiedenen Ereignisdetails finden Sie unter [Ereignisse zur Änderung des Status der Instance-Flotte](#).

Beispiele für Bereitstellungs-Timeouts bei der Clustergrößenänderung

Geben Sie ein Bereitstellungs-Timeout für die Größenänderung mit dem AWS CLI an

Im folgenden Beispiel wird der `create-cluster`-Befehl verwendet, um ein Bereitstellungs-Timeout für die Größenänderung hinzuzufügen.

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-XXXXX"]}' \
--instance-fleets
' [{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceType":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
- 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}}, {"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
- 2"}] ]'
```

Im folgenden Beispiel wird der `modify-instance-fleet`-Befehl verwendet, um ein Bereitstellungs-Timeout für die Größenänderung hinzuzufügen.

```
aws emr modify-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet '{"InstanceFleetId":"if-XXXXXXXXXXXX","ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":60}}}' \
--region us-east-1
```

Im folgenden Beispiel wird der `add-instance-fleet-command` verwendet, um ein Bereitstellungs-Timeout für die Größenänderung hinzuzufügen.

```
aws emr add-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet
'{"InstanceFleetType":"TASK","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTypeCo
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":35}}}' \
--region us-east-1
```

Geben Sie ein Bereitstellungs-Timeout für die Größenänderung und den Start mit dem AWS CLI

Im folgenden Beispiel wird der `create-cluster-Befehl` verwendet, um ein Bereitstellungs-Timeout für die Größenänderung hinzuzufügen.

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
'[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSpeci
{"OnDemandSpecification":{"AllocationStrategy":"lowest-price"}}, {"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
- 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}}, {"ResizeSpecifications":
```



```
{
  "SpotResizeSpecification": {
    "TimeoutDurationMinutes": 20
  },
  "OnDemandResizeSpecification": {
    "TimeoutDurationMinutes": 25
  },
  "InstanceTypeConfigs": [
    {
      "WeightedCapacity": 1,
      "EbsConfiguration": {
        "EbsBlockDeviceConfigs": [
          {
            "VolumeSpecification": {
              "SizeInGB": 32,
              "VolumeType": "gp2"
            },
            "VolumesPerInstance": 2
          }
        ]
      },
      "BidPriceAsPercentageOfOnDemandPrice": 2
    }
  ]
}
```

Überlegungen zur Größenänderung von Bereitstellungs-Timeouts

Wenn Sie Timeouts für die Cluster-Bereitstellung für Ihre Instanceflotten konfigurieren, sollten Sie die folgenden Verhaltensweisen berücksichtigen.

- Sie können Bereitstellungs-Timeouts sowohl für Spot Instances als auch für On-Demand-Instances konfigurieren. Das Mindestzeitlimit für die Bereitstellung beträgt 5 Minuten. Das maximale Bereitstellungszeitlimit beträgt 7 Tage.
- Sie können Bereitstellungs-Zeitlimit nur für einen EMR-Cluster konfigurieren, der Instance-Flotten verwendet. Sie müssen jeden Core und jede Aufgaben-Flotte separat konfigurieren.
- Wenn Sie einen Cluster erstellen, können Sie Bereitstellungs-Zeitlimits konfigurieren. Sie können ein Zeitlimit für einen laufenden Cluster hinzufügen oder ein vorhandenes Zeitlimit aktualisieren.
- Wenn Sie mehrere Größenänderungsvorgänge einreichen, verfolgt Amazon EMR die Bereitstellungs-Timeouts für jeden Größenänderungsvorgang. Legen Sie beispielsweise das Bereitstellungs-Timeout für einen Cluster auf Minuten fest. **60** Senden Sie dann nach und nach einen Vorgang **R1** zur Größenänderung. **T1** Reichen Sie jeweils einen zweiten Vorgang **R2** zur Größenänderung ein. **T2** Das Bereitstellungs-Timeout für R1 läuft am ab. **T1 + 60 minutes** Das Bereitstellungs-Timeout für R2 läuft am ab. **T2 + 60 minutes**
- Wenn Sie vor Ablauf des Timeouts einen neuen Vorgang zur Skalierung der Größe einreichen, versucht Amazon EMR weiterhin, Kapazität für Ihren EMR-Cluster bereitzustellen.

Cluster-Scale-Down-Optionen für Amazon EMR-Cluster

Note

Optionen für das Herunterskalierungs-Verhalten werden seit der Amazon-EMR-Version 5.10.0 nicht mehr unterstützt. Aufgrund der Einführung der Abrechnung pro Sekunde in Amazon EC2 wird das Standard-Scale-Down-Verhalten für Amazon EMR-Cluster jetzt bei Abschluss der Aufgabe beendet.

In den Amazon EMR-Versionen 5.1.0 bis 5.9.1 gibt es zwei Optionen für das Scale-Down-Verhalten: Beenden innerhalb der Instanz-Stunden-Grenze für EC2 Amazon-Fakturierung oder Beenden bei Abschluss der Aufgabe. Ab Amazon EMR Version 5.10.0 ist die Einstellung für die Kündigung innerhalb der Instance-Stundengrenze veraltet, da Amazon die Abrechnung pro Sekunde eingeführt hat. EC2 Wir raten davon ab, die Beendigung zur Instance-Stundengrenze zu verwenden, wenn diese Option angeboten wird.

Warning

Wenn Sie die Option „AWS CLI `amodify-instance-groups`“ verwenden `EC2InstanceIdsToTerminate`, werden diese Instances sofort beendet, ohne Berücksichtigung dieser Einstellungen und unabhängig vom Status der Anwendungen, die auf ihnen ausgeführt werden. Wenn eine Instance auf diese Weise beendet wird, besteht das Risiko von Datenverlusten und unvorhersehbarem Clusterverhalten.

Wenn „Bei Abschluss der Aufgabe beenden“ angegeben ist, lehnt Amazon EMR Aufgaben ab und entfernt Aufgaben von Knoten, bevor die Amazon-Instances beendet werden. EC2 Wenn eines der beiden Verhaltensweisen angegeben ist, beendet Amazon EMR keine EC2 Amazon-Instances in Core-Instance-Gruppen, wenn dies zu einer Beschädigung von HDFS führen könnte.

Beendigung bei Aufgaben-Abschluss

Mit Amazon EMR können Sie eine Herunterskalierung für den Cluster durchführen, ohne dass es zu Auswirkungen auf den Workload kommt. Amazon EMR versucht, YARN, HDFS und andere Daemons auf Kern- und Taskknoten während einer Größenänderung ordnungsgemäß außer Betrieb zu nehmen, ohne Daten zu verlieren oder Jobs zu unterbrechen. Amazon EMR verkleinert Instance-Gruppen nur, wenn die zu den Gruppen zugewiesenen Aufgaben abgeschlossen sind und sie unausgelastet sind. Bei YARN NodeManager Graceful Decommission können Sie die Zeit, die ein Knoten auf die Außerbetriebnahme wartet, manuell anpassen.

Note

Bei einer ordnungsgemäßen Außerbetriebnahme kann es zu Datenverlust kommen. Stellen Sie sicher, dass Sie Ihre Daten sichern.

 Important

Es ist möglich, dass HDFS-Daten beim ordnungsgemäßen Austausch einer fehlerhaften Core-Instance dauerhaft verloren gehen. Wir empfehlen, dass Sie Ihre Daten immer sichern.

Die Dauer wird mit einer Eigenschaft in der YARN-site-Konfigurationsklassifizierung eingerichtet. Wenn Sie Amazon-EMR-Version 5.12.0 und höher verwenden, geben Sie die Eigenschaft `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` an. Bei Verwendung von früheren Amazon-EMR-Versionen geben Sie die Eigenschaft `YARN.resourcemanager.decommissioning.timeout` an.

Wenn nach diesem Stilllegungszeitraum noch Container oder YARN-Anwendungen ausgeführt werden, wird die Stilllegung des Knotens erzwungen. YARN plant betroffene Container in anderen Knoten neu. Die Standardwert ist 3600 Sekunden (eine Stunde). Sie können den Timeout auf einen extrem hohen Wert festlegen, um die ordnungsgemäße Verkleinerung zu verzögern. Weitere Informationen finden Sie unter [Ordnungsgemäßes Stilllegen von YARN Knoten](#) in der Apache-Hadoop-Dokumentation.

Aufgabenknoten-Gruppen

Amazon EMR wählt in intelligenter Weise Instances aus, die keine Aufgaben im Zusammenhang mit einem Schritt oder einer Anwendung ausführen und entfernt diese Instance zuerst aus einem Cluster. Wenn alle Instances im Cluster genutzt werden, wartet Amazon EMR auf den Abschluss von Tasks in einer Instance, bevor diese aus dem Cluster entfernt wird. Die standardmäßige Leerlaufzeit beträgt eine Stunde. Dieser Wert kann mit der Einstellung `YARN.resourcemanager.decommissioning.timeout` geändert werden. Amazon EMR nutzt die neue Einstellung dynamisch. Sie können diesen Wert auf eine beliebig große Zahl festlegen, um sicherzustellen, dass Amazon EMR keine Aufgaben beendet und gleichzeitig die Clustergröße reduziert.

Core-Knoten-Gruppen

Auf Kernknoten müssen sowohl YARN NodeManager - als auch DataNode HDFS-Daemons außer Betrieb genommen werden, damit die Instanzgruppe reduziert werden kann. Bei YARN stellt das ordnungsgemäße Verkleinern sicher, dass ein für die Außerbetriebnahme vorgesehener Knoten nur dann in den Status DECOMMISSIONED wechselt, wenn es keine ausstehenden oder unvollständigen Container oder Anwendungen gibt. Die Außerbetriebnahme wird direkt beendet, falls es zu Beginn der Außerbetriebnahme keine laufenden Container auf dem Knoten gibt.

Bei HDFS stellt die ordnungsgemäße Verkleinerung sicher, dass die Zielkapazität von HDFS für alle vorhandenen Blöcke ausreichend ist. Wenn die Zielkapazität nicht groß genug ist, wird nur ein Teil der Core-Instances außer Betrieb gestellt. So können die verbleibenden Knoten die aktuell in HDFS vorhandenen Daten verarbeiten. Stellen Sie für eine weitere Außerbetriebnahme zusätzliche HDFS-Kapazität sicher. Sie sollten auch versuchen, den Schreibaufwand zu minimieren, I/O bevor Sie versuchen, Instanzgruppen zu reduzieren. Übermäßiger Schreibvorgang I/O kann den Abschluss des Größenänderungsvorgangs verzögern.

Ein weiterer Faktor ist der Standard-Replikationsfaktor (`dfs.replication`) in `/etc/hadoop/conf/hdfs-site`. Amazon EMR konfiguriert den Wert basierend auf der Anzahl der Instances im Cluster: 1 bei 1–3 Instances, 2 für Cluster mit 4–9 Instances und 3 für Cluster mit mehr als zehn Instances.

 Warning

1. Das Festlegen von `dfs.replication` auf 1 auf Clustern mit weniger als vier Knoten kann zu einem HDFS-Datenverlust führen, wenn ein einzelner Knoten ausfällt. Wir empfehlen, für Produktionsworkloads einen Cluster mit mindestens vier Core-Knoten zu verwenden.
2. Amazon EMR erlaubt Clustern nicht, Core-Knoten unter `dfs.replication` zu skalieren. Bei `dfs.replication = 2` z. B. beträgt die Mindestanzahl von Core-Knoten 2.
3. Wenn Sie verwaltete Skalierung oder Auto-Scaling verwenden oder die Größe Ihres Clusters manuell ändern möchten, empfehlen wir Ihnen, `dfs.replication` auf 2 oder höher einzustellen.

Durch die schrittweise Reduzierung können Sie die Anzahl der Core-Knoten nicht unter den HDFS-Replikationsfaktor reduzieren. Auf diese Weise kann HDFS Dateien aufgrund unzureichender Replikate schließen. Um dieses Limit zu umgehen, verringern Sie den Replikationsfaktor und starten Sie den NameNode Daemon neu.

Das Herunterskalierungs-Verhalten für Amazon-EMR-Cluster konfigurieren

Note

Die Option zum Herunterskalieren zur Instance-Stunde wird für Amazon EMR Version 5.10.0 und höher nicht mehr unterstützt. Die folgenden Optionen für das Scale-Down-Verhalten werden nur in der Amazon-EMR-Konsole für die Versionen 5.1.0 bis 5.9.1 angezeigt.

Sie können die AWS Management Console, oder die Amazon EMR-API verwenden AWS CLI, um das Scale-Down-Verhalten bei der Erstellung eines Clusters zu konfigurieren.

Console

Um das Scale-Down-Verhalten mit der Konsole zu konfigurieren

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters und dann Create cluster aus.
3. Wählen Sie im Abschnitt Optionen für Clusterskalierung und -bereitstellung die Option Benutzerdefinierte automatische Skalierung verwenden aus. Wählen Sie unter Benutzerdefinierte automatische Skalierungsrichtlinien die Plus-Aktionsschaltfläche aus, um Skalierungsrichtlinien hinzuzufügen. Wir empfehlen, dass Sie sowohl Richtlinien für die Skalierung als auch für die horizontale Skalierung hinzufügen. Wenn Sie nur einen Satz von Richtlinien hinzufügen, führt Amazon EMR nur eine unidirektionale Skalierung durch und Sie müssen die anderen Aktionen manuell ausführen.
4. Wählen Sie alle anderen Optionen aus, die für Ihren Cluster gelten.
5. Um Ihren Cluster jetzt zu starten, wählen Sie Cluster erstellen aus.

AWS CLI

Um das Scale-Down-Verhalten zu konfigurieren, verwenden Sie AWS CLI

- Verwenden Sie für die `--scale-down-behavior`-Option entweder `TERMINATE_AT_INSTANCE_HOUR` oder `TERMINATE_AT_TASK_COMPLETION`.

Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“

In diesem Abschnitt werden die Methoden zum Beenden eines Clusters beschrieben. Informationen zum Aktivieren des Beendigungsschutzes und zum automatischen Beenden von Clustern finden Sie unter [Steuern Sie die Kündigung des Amazon EMR-Clusters](#). Sie können Cluster mit dem Status STARTING, RUNNING oder WAITING beenden. Ein Cluster mit dem Status WAITING muss beendet werden. Andernfalls wird er unbegrenzt ausgeführt, und verursacht Gebühren für Ihr Konto. Sie können einen Cluster beenden, der den Status STARTING nicht verlässt oder der einen bestimmten Schritt nicht durchführen kann.

Wenn Sie einen Cluster beenden, bei dem der Beendigungsschutz aktiviert ist, müssen Sie den Beendigungsschutz deaktivieren, bevor Sie den Cluster beenden können. Cluster können mithilfe der Konsole, der oder programmgesteuert mithilfe der AWS CLI API beendet werden.

`TerminateJobFlows`

Je nach Konfiguration des Clusters kann es zwischen 5 und 20 Minuten dauern, bis der Cluster die zugewiesenen Ressourcen, wie EC2 z. B. Instanzen, vollständig beendet und freigegeben hat.

Note

Sie können einen beendeten Cluster nicht neu starten, aber Sie können einen beendeten Cluster klonen, um seine Konfiguration für einen neuen Cluster wiederzuverwenden. Weitere Informationen finden Sie unter [Klonen Sie einen Amazon EMR-Cluster mithilfe der Konsole](#).

Important

Amazon EMR verwendet die [Amazon EMR-Servicerolle und die AWSServiceRoleForEMRCleanup Rolle](#), um Clusterressourcen in Ihrem Konto zu bereinigen, die Sie nicht mehr verwenden, z. B. EC2 Amazon-Instanzen. Sie müssen Aktionen für die Rollenrichtlinien angeben, um die Ressourcen zu löschen oder zu beenden. Andernfalls kann Amazon EMR diese Bereinigungsaktionen nicht durchführen, und es können Kosten für ungenutzte Ressourcen anfallen, die im Cluster verbleiben.

Einen Cluster mit der Konsole zu beenden

Sie können einen oder mehrere Cluster mithilfe der Amazon-EMR-Konsole beenden. Die Schritte zum Beenden eines Clusters über die Konsole variieren je nachdem, ob der Beendigungsschutz aktiviert oder deaktiviert ist. Um einen geschützten Cluster zu beenden, müssen Sie zuerst den Beendigungsschutz deaktivieren.

Console

Um einen Cluster mit der Konsole zu beenden

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie Clusters und dann den Cluster aus, den Sie beenden möchten.
3. Wählen Sie im Dropdownmenü Aktionen die Option Cluster beenden aus, um die Aufforderung Cluster beenden zu öffnen.
4. Wählen Sie an der Eingabeaufforderung die Option Beenden. Je nach Clusterkonfiguration kann die Kündigung 5 bis 10 Minuten dauern. Weitere Informationen zum Amazon-EMR-Cluster finden Sie unter [Beenden Sie einen Amazon EMR-Cluster im Status „Start“, „Wird ausgeführt“ oder „Wartend“](#).

Beenden eines Clusters mithilfe der AWS CLI

Um einen ungeschützten Cluster mit dem AWS CLI

Um einen ungeschützten Cluster mit dem zu beenden AWS CLI, verwenden Sie den `terminate-clusters` Unterbefehl mit dem Parameter `--cluster-ids`.

- Geben Sie den folgenden Befehl ein, um einen einzelnen Cluster zu beenden und ihn durch Ihre Cluster-ID zu ersetzen `j-3KVXXXXXXXX7UG`.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXXX7UG
```

Um mehrere Cluster zu beenden, geben Sie den folgenden Befehl ein und ersetzen Sie `j-3KVXXXXXXXX7UG` und `j-WJ2XXXXXXXX8EU` durch Ihren Cluster IDs.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXXXX7UG j-WJ2XXXXXXXX8EU
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Um einen geschützten Cluster mit dem zu beenden AWS CLI

Um einen geschützten Cluster mit dem zu beenden AWS CLI, deaktivieren Sie zunächst den Kündigungsschutz mithilfe des `modify-cluster-attributes` Unterbefehls mit dem `--no-termination-protected` Parameter. Verwenden Sie dann den Unterbefehl `terminate-clusters` mit dem Parameter `--cluster-ids`, um den Cluster zu beenden.

1. Geben Sie den folgenden Befehl ein, um den Kündigungsschutz zu deaktivieren, und `j-3KVTXXXXXX7UG` ersetzen Sie ihn durch Ihre Cluster-ID.

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
```

2. Um den Cluster zu beenden, geben Sie den folgenden Befehl ein und `j-3KVXXXXXX7UG` ersetzen Sie ihn durch Ihre Cluster-ID.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXX7UG
```

Um mehrere Cluster zu beenden, geben Sie den folgenden Befehl ein und ersetzen Sie `j-3KVXXXXXX7UG` und `j-WJ2XXXXXX8EU` durch Ihren Cluster IDs.

```
aws emr terminate-clusters --cluster-ids j-3KVXXXXXX7UG j-WJ2XXXXXX8EU
```

Weitere Informationen zur Verwendung von Amazon EMR-Befehlen finden Sie AWS CLI unter <https://docs.aws.amazon.com/cli/latest/reference/emr>.

Beenden eines Clusters mit der API

Der `TerminateJobFlows` Vorgang beendet die Schrittverarbeitung, lädt alle Protokolldaten von Amazon EC2 auf Amazon S3 hoch (falls konfiguriert) und beendet den Hadoop-Cluster. Ein Cluster wird außerdem automatisch beendet, wenn Sie in einer `KeepJobAliveWhenNoSteps`-Anforderung `False` auf `RunJobFlows` festlegen.

Sie können diese Aktion verwenden, um entweder einen einzelnen Cluster oder eine Liste von Clustern nach ihrem Cluster zu beenden. IDs

Weitere Hinweise zu den Eingabeparametern, die nur für `terminateJobFlows` gelten, finden Sie unter [TerminateJobFlows](#). Weitere Informationen zu den grundlegenden Parametern in der Anfrage finden Sie unter [Allgemeine Anforderungsparameter](#).

Klonen Sie einen Amazon EMR-Cluster mithilfe der Konsole

Sie können die Amazon-EMR-Konsole zum Klonen eines Clusters verwenden, wodurch eine Kopie der Konfiguration des ursprünglichen Cluster als Basis für einen neuen Cluster erstellt wird.

Console

Um einen Cluster mit der Konsole zu klonen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon EMR-Konsole unter <https://console.aws.amazon.com/emr>.
2. Wählen Sie EC2 im linken Navigationsbereich unter EMR on die Option Clusters aus.
3. Um einen Cluster aus der Cluster-LIS zu klonen
 - a. Verwenden Sie die Such- und Filteroptionen, um den Cluster, den Sie klonen möchten, in der Listenansicht zu finden.
 - b. Markieren Sie das Kontrollkästchen links neben der Zeile für den Cluster, den Sie klonen möchten.
 - c. Die Option Klonen ist jetzt oben in der Listenansicht verfügbar. Wählen Sie Klonen aus, um den Klonvorgang zu starten. Wenn für den Cluster Schritte konfiguriert sind, wählen Sie Schritte einschließen und Weiter aus, wenn Sie die Schritte zusammen mit den anderen Clusterkonfigurationen klonen möchten.
 - d. Überprüfen Sie die Einstellungen für den neuen Cluster, die aus dem geklonten Cluster kopiert wurden. Passen Sie die Einstellungen bei Bedarf an. Wenn Sie mit der Konfiguration des neuen Clusters zufrieden sind, wählen Sie Cluster erstellen aus, um den neuen Cluster zu starten.
4. Wie Sie einen Cluster von einer Cluster-Detailseite aus klonen
 - a. Um zur Detailseite des Clusters zu gelangen, den Sie klonen möchten, wählen Sie dessen Cluster-ID aus der Cluster-Listenansicht aus.

- b. Wählen Sie oben auf der Cluster-Detailseite im Menü Aktionen die Option Cluster klonen aus, um den Klonvorgang zu starten. Wenn für den Cluster Schritte konfiguriert sind, wählen Sie Schritte einschließen und Weiter aus, wenn Sie die Schritte zusammen mit den anderen Clusterkonfigurationen klonen möchten.
- c. Überprüfen Sie die Einstellungen für den neuen Cluster, die aus dem geklonten Cluster kopiert wurden. Passen Sie die Einstellungen bei Bedarf an. Wenn Sie mit der Konfiguration des neuen Clusters zufrieden sind, wählen Sie Cluster erstellen aus, um den neuen Cluster zu starten.

Automatisieren Sie wiederkehrende Amazon EMR-Cluster mit AWS Data Pipeline

AWS Data Pipeline ist ein Dienst, der die Übertragung und Transformation von Daten automatisiert. Sie können ihn verwenden, um Eingabedaten in Amazon S3 zu verlagern und das Starten von Clustern zu planen, die diese Daten verarbeiten. Betrachten wir zum Beispiel den Fall, bei dem ein Webserver Datenverkehrsprotokolle aufzeichnet. Wenn Sie einen wöchentlichen Cluster zur Analyse der Verkehrsdaten ausführen möchten, können Sie ihn AWS Data Pipeline zur Planung dieser Cluster verwenden. AWS Data Pipeline ist ein datengesteuerter Workflow, sodass eine Aufgabe (Starten des Clusters) von einer anderen Aufgabe (Verschieben der Eingabedaten nach Amazon S3) abhängig sein kann. Der Workflow verfügt außerdem über eine robuste Wiederholungsfunktionalität.

Weitere Informationen zu AWS Data Pipeline finden Sie im [AWS Data Pipeline Entwicklerhandbuch](#), insbesondere in den Tutorials zu Amazon EMR:

- [Tutorial: Einen Amazon-EMR-Auftragsverlauf starten](#)
- [Erste Schritte: Webprotokolle mit AWS Data Pipeline Amazon EMR und Hive verarbeiten](#)
- [Tutorial: Amazon DynamoDB importieren und exportieren mit AWS Data Pipeline](#)

Amazon EMR-Tutorials

Ein EMR-Cluster läuft in einem komplexen Ökosystem. Das folgende Tutorial behandelt wichtige Anwendungsfälle.

Themen

- [Amazon EMR on EC2 — Verbesserte Überwachung mit CloudWatch benutzerdefinierten Metriken und Protokollen](#)

Amazon EMR on EC2 — Verbesserte Überwachung mit CloudWatch benutzerdefinierten Metriken und Protokollen

Übersicht

Amazon EMR bietet leistungsstarke, kostengünstige Funktionen zur Verarbeitung großer Datenmengen. Um Leistung und Ressourcennutzung zu maximieren, ist eine effektive Überwachung unerlässlich. Amazon CloudWatch bietet umfassende Beobachtbarkeit für EMR-Cluster, sodass Sie Metriken und Protokolle in Echtzeit verfolgen können. In diesem Dokument wird beschrieben, wie Sie:

1. Konfigurieren Sie den CloudWatch Agenten zum Senden von EC2 EMR-On-Protokollen an CloudWatch
2. Fügen Sie mithilfe von Klassifizierungen benutzerdefinierte Hadoop-, YARN- und HBase Metriken hinzu
3. Überwachen Sie Metriken über integrierte Dashboards
4. Verfolgen Sie Cluster-Logs über CloudWatch Protokollgruppen

Voraussetzungen und Hintergrund

Standardmäßig sendet Amazon EMR ohne zusätzliche Kosten CloudWatch alle fünf Minuten grundlegende Metriken. Mit EMR Version 7.0+ können Sie den CloudWatch Agenten einsetzen für:

- Erfassen Sie 34 zusätzliche detaillierte Messwerte in Intervallen von einer Minute (es fallen zusätzliche Gebühren an)
- Sammeln Sie Metriken von allen Clusterknoten

- Aggregieren Sie Daten auf dem primären Knoten, bevor Sie sie an senden CloudWatch
- Greifen Sie über die Registerkarte Monitoring oder die Konsole der EMR-Konsole auf Metriken zu CloudWatch

EMR 7.1 erweitert diese Funktionen, sodass Sie den Agenten so konfigurieren können, dass er spezielle Metriken aus Hadoop, YARN und HBase Komponenten erfasst. Für Umgebungen, in denen Prometheus verwendet wird, können Metriken an Amazon Managed Service for Prometheus weitergeleitet werden.

CloudWatch Agentenkonfiguration für Protokolle

Um EMR-Logs zu erfassen CloudWatch, erstellen Sie eine `cloudwatch-config.json`-Datei, die definiert, welche Protokolldateien gesammelt werden sollen:

`cloudwatch-config.json`

```
{
  "agent": {"metrics_collection_interval":60,"logfile":"/var/log/
emr-cluster-metrics/amazon-cloudwatch-agent/amazon-cloudwatch-
agent.log","run_as_user":"****","omit_hostname":true},
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/mnt/var/log/hadoop-yarn/hadoop-yarn-resourcemanager-*",
            "log_group_name": "/emr/yarn/resourcemnager",
            "log_stream_name": "{instance_id}",
            "publish_multi_logs" : true
          },
          {
            "file_path": "/var/log/hadoop-hdfs/hadoop-hdfs-namenode-*",
            "log_group_name": "/emr/hdfs/namenode",
            "log_stream_name": "{instance_id}",
            "publish_multi_logs" : true
          }
        ]
      }
    }
  }
}
```

Bootstrap-Skript für die Agentenkonfiguration CloudWatch

Um Ihre benutzerdefinierte CloudWatch Konfiguration auf EMR-Knoten anzuwenden, erstellen Sie ein Bootstrap-Skript, das den CloudWatch Agenten mit Ihren Einstellungen neu startet. Dieses Skript stellt sicher, dass der Agent nach der Cluster-Bereitstellung mit Ihren spezifischen Protokollerfassungsparametern ausgeführt wird.

Das Bootstrap-Skript erstellen

Erstellen Sie eine Datei namens `cloudwatch-agent-bootstrap.sh` mit dem folgenden Inhalt:

```
#!/bin/bash
set -xe

EMR_SECONDARY_BA_SCRIPT=$(cat << 'EOF'
while true; do
NODEPROVISIONSTATE=$(sed -n '/localInstance [{}/,/[]]/ {/nodeProvisionCheckinRecord
[{}]/,/[]]/ {/status:/ p}}' /emr/instance-controller/lib/info/job-flow-state.txt | awk
'{ print $2 }')

if [ "$NODEPROVISIONSTATE" == "SUCCESSFUL" ]; then
sleep 10
echo "Running my post provision bootstrap"
NODETYPE=$(cat /mnt/var/lib/instance-controller/extraInstanceData.json | jq -r
'.instanceRole' | awk '{print tolower($0)}')

# Copy config file on the instance
sudo aws s3 cp s3://amzn-s3-demo-bucket1/cloudwatch-config.json /etc/emr-cluster-
metrics/amazon-cloudwatch-agent/conf/emr-amazon-cloudwatch-agent.json

# Stop the current agent
sudo /usr/bin/amazon-cloudwatch-agent-ctl -a stop

# Start the agent with the created config file
sudo /usr/bin/amazon-cloudwatch-agent-ctl -a fetch-config -s -m ec2 -c file:/etc/emr-
cluster-metrics/amazon-cloudwatch-agent/conf/emr-amazon-cloudwatch-agent.json

# Status CW Agent
echo "Status CW Agent"
sudo /usr/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status

exit
fi
```

```
sleep 10
done
EOF
)

echo "${EMR_SECONDARY_BA_SCRIPT}" | tee -a /tmp/emr-secondary-ba.sh
chmod u+x /tmp/emr-secondary-ba.sh
/tmp/emr-secondary-ba.sh > /tmp/emr-secondary-ba.log 2>&1 &
exit 0
```

Wichtiger Hinweis zur Konfiguration

Important

Bevor Sie das Skript hochladen, **<amzn-s3-demo-bucket1>** ersetzen Sie es durch den tatsächlichen Namen Ihres S3-Buckets, in dem Sie die Datei cloudwatch-config.json aus dem vorherigen Schritt gespeichert haben. Dadurch wird sichergestellt, dass das Bootstrap-Skript Ihre Konfigurationsdatei während der Cluster-Initialisierung abrufen kann.

Dieses Bootstrap-Skript wird:

- Warten Sie, bis die Knotenbereitstellung abgeschlossen ist
- Laden Sie Ihre benutzerdefinierte Konfiguration CloudWatch herunter
- Stoppen Sie jeden laufenden CloudWatch Agenten
- Starten Sie den Agenten mit Ihrer spezifischen Konfiguration neu
- Protokollieren Sie den Status des Agenten zur Fehlerbehebung

Benutzerdefinierte metrische Klassifizierungen für Hadoop, YARN und HBase

Zusätzlich zu den CloudWatch Standardmetriken können Sie Ihre Überwachungsfunktionen verbessern, indem Sie benutzerdefinierte anwendungsspezifische Metriken für Ihre EMR-Clusterkomponenten konfigurieren. Die Konfigurations-API von Amazon EMR bietet eine flexible Möglichkeit, genau zu definieren, welche Metriken Sie sammeln möchten.

Konfiguration benutzerdefinierter Metriken

Sie können die Erfassung benutzerdefinierter Metriken auf zwei Arten implementieren:

- Während der Clustererstellung für neue Cluster
- Als Rekonfiguration für bestehende Cluster über die EMR-Konsole

Eine Klassifizierungsdatei erstellen

Die Klassifizierungsdatei definiert, welche spezifischen Komponentenmetriken aus Ihrem Cluster gesammelt werden sollen. Im Folgenden finden Sie eine Beispielstruktur für die Erfassung benutzerdefinierter Hadoop-Metriken:

```
[
  {
    "Classification": "emr-metrics",
    "Configurations": [
      {
        "Classification": "emr-hadoop-hdfs-datanode-metrics",
        "Properties": {
          "Hadoop:service=DataNode,name=DataNodeActivity-*":
"DatanodeNetworkErrors,TotalReadTime,TotalWriteTime,BytesRead,BytesWritten,RemoteBytesRead,RemoteBytesWritten",
          "otel.metric.export.interval": "30000"
        }
      },
      {
        "Classification": "emr-hadoop-yarn-nodemanager-metrics",
        "Properties": {
          "Hadoop:service=NodeManager,name=JvmMetrics":
"MemNonHeapUsedM,MemNonHeapCommittedM,MemNonHeapMaxM,MemHeapUsedM,MemHeapCommittedM,MemHeapMaxM",
          "Hadoop:service=NodeManager,name=NodeManagerMetrics":
"ContainerCpuUtilization,NodeCpuUtilization,ContainersCompleted,ContainersFailed,ContainersKilled",
          "otel.metric.export.interval": "20000"
        }
      }
    ],
    "Properties": {}
  }
]
```

Implementierungsschritte

1. Erstellen Sie eine JSON-Datei mit den gewünschten metrischen Klassifizierungen.
2. Passen Sie die Metriken an Ihre Überwachungsanforderungen an.
3. Speichern Sie die Datei und laden Sie sie in Ihren S3-Bucket hoch.

4. Verweisen Sie auf diese Datei, wenn Sie einen neuen Cluster erstellen oder einen vorhandenen neu konfigurieren.

Bewährte Methoden

- Erfassen Sie nur Metriken, die aussagekräftige Erkenntnisse für Ihre Workloads liefern.
- Berücksichtigen Sie das Intervall für die Erfassung von Metriken auf der Grundlage Ihrer Überwachungsanforderungen.
- Die vollständige Liste der verfügbaren Metriken für jede Komponente finden Sie in der AWS Dokumentation.
- Gruppieren Sie verwandte Metriken innerhalb derselben Klassifizierung, um eine bessere Organisation zu erreichen.

Dieser Ansatz ermöglicht es Ihnen, Ihre Überwachung auf die kritischsten Metriken für Ihre spezifischen EMR-Anwendungen zu konzentrieren und so einen tieferen Einblick in die Cluster-Leistung zu erhalten.

Bereitstellung eines EMR-Clusters mit Integration CloudWatch

Gehen Sie wie folgt vor, um einen Amazon EMR-Cluster zu erstellen, der automatisch Protokolle und benutzerdefinierte Metriken sendet an: CloudWatch

Schritt 1: Aktivieren Sie den Agenten CloudWatch

Gehen Sie beim Erstellen Ihres EMR-Clusters über die AWS Management Console wie folgt vor:

1. Navigieren Sie während der Clustererstellung zum Abschnitt Anwendungen.
2. Wählen Sie die Kontrollkästchen für Ihre primären Anwendungen (Hadoop, Spark usw.) aus.
3. Scrollen Sie, um die Option Amazon CloudWatch Agent zu finden und auszuwählen.
4. Dadurch wird der Agent in Ihrem Cluster aktiviert, was für die Erfassung erweiterter Metriken und Protokolle unerlässlich ist.

Der CloudWatch Agent wird auf allen Knoten in Ihrem Cluster installiert, sodass er System- und Anwendungsmetriken in den konfigurierten Intervallen sammeln kann.

▼

Name and applications - required [Info](#)

Name your cluster and choose the applications that you want to install to your cluster.

Name

TestCluster

Amazon EMR release [Info](#)

A release contains a set of applications which can be installed on your cluster.

emr-7.7.0 ▼

Application bundle

Spark Interactive



Core Hadoop



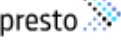
Flink



HBase



Presto



Trino



Custom



☒ AmazonCloudWatchAgent 1.300032.2

☐ HCatalog 3.1.3
 ☐ Hue 4.11.0
 ☒ Livy 0.8.0
 ☐ Pig 0.17.0
 ☐ TensorFlow 2.16.1
 ☐ Zeppelin 0.11.1

☐ Flink 1.20.0

☒ Hadoop 3.4.0
 ☒ JupyterEnterpriseGateway 2.6.0
 ☐ Oozie 5.2.1
 ☐ Presto 0.287
 ☒ Tez 0.10.2
 ☐ ZooKeeper 3.9.2

☒ HBase 2.6.1

☒ Hive 3.1.3
 ☐ JupyterHub 1.5.0
 ☐ Phoenix 5.2.1
 ☒ Spark 3.5.3
 ☐ Trino 467

Einen Cluster erstellen und die verfügbaren Bundles anzeigen.

Note

Der CloudWatch Agent ist in EMR Version 7.0 und höher verfügbar. Die Aktivierung dieser Komponente ist für die Erfassung und Weiterleitung von benutzerdefinierten Messwerten und die in diesem Handbuch beschriebene Protokollweiterleitung erforderlich.

Schritt 2: Fügen Sie die Bootstrap-Aktion für die Protokollerfassung hinzu

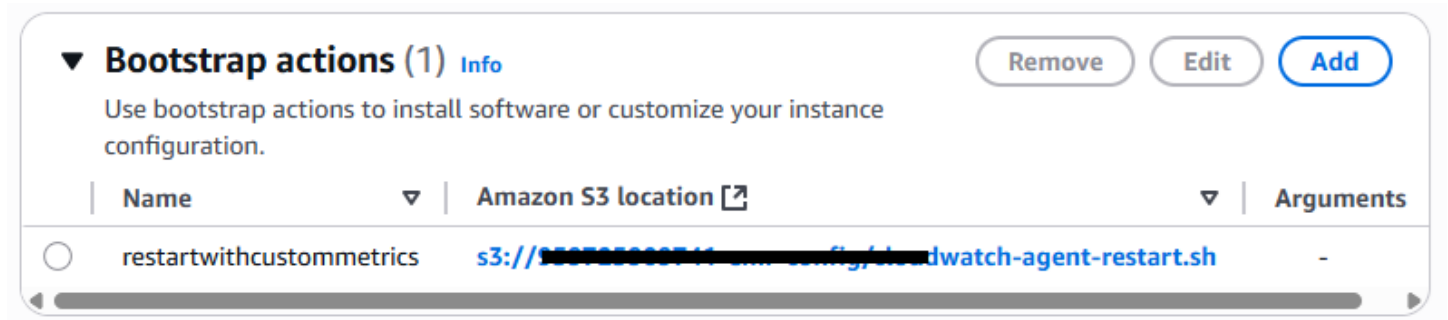
So konfigurieren Sie den CloudWatch Agenten so, dass er bestimmte Protokolldateien sammelt und weiterleitet an CloudWatch:

1. Navigieren Sie im Assistenten zum Erstellen von EMR-Clustern zum Abschnitt Bootstrap-Aktionen

2. Klicken Sie auf Bootstrap-Aktion hinzufügen
3. Wählen Sie im Drop-down-Menü die Option Benutzerdefinierte Aktion aus
4. Geben Sie einen Namen für Ihre Bootstrap-Aktion ein (z. B. Configure CloudWatch Agent)
5. Geben Sie im Feld Skriptspeicherort den S3-Pfad zu Ihrem cloudwatch-agent-bootstrap .sh-Skript ein (z. B. s3://your-bucket-name/cloudwatch-agent-bootstrap.sh)
6. Klicken Sie auf Hinzufügen, um die Bootstrap-Aktion zu speichern

Diese Bootstrap-Aktion wird beim Start des Clusters ausgeführt. Dabei wird sichergestellt, dass der ordnungsgemäß mit Ihren benutzerdefinierten Einstellungen konfiguriert CloudWatchagent ist, um die in Ihrer Konfigurationsdatei angegebenen Protokolldateien zu sammeln und weiterzuleiten.

Der Agent beginnt automatisch mit der Erfassung von Protokollen, sobald die Knoten bereitgestellt sind, sodass Sie mithilfe CloudWatch von Protokollen nahezu in Echtzeit Einblick in Ihre Clustervorgänge erhalten.



Verwendung von Bootstrap-Aktionen.

Schritt 3: Konfigurieren Sie die Erfassung benutzerdefinierter Metriken

So aktivieren Sie die Erfassung von benutzerdefinierten Hadoop-, YARN- oder HBase Metriken, die über den Standardsatz hinausgehen:

1. Navigieren Sie im Assistenten zum Erstellen von EMR-Clustern zum Abschnitt Konfigurationen.
2. Klicken Sie auf die Schaltfläche Konfigurationen bearbeiten, um die Konfigurationsoptionen zu erweitern.
3. Wählen Sie in der Dropdownliste für die Konfigurationsmethode die Option JSON aus Amazon S3 laden aus.
4. Geben Sie den S3-URI-Pfad zu Ihrer Klassifizierungsdatei für benutzerdefinierte Metriken ein (z. B. s3://amzn-s3-demo-bucket1/emr-metrics-classification.json).

5. Klicken Sie auf Laden, um die Konfiguration zu analysieren.
6. Stellen Sie sicher, dass die Konfiguration in der Konsolenoberfläche korrekt angezeigt wird.
7. Klicken Sie auf Änderungen speichern, um diese Metrikkonfigurationen auf Ihren Cluster anzuwenden.

In diesem Schritt wird der CloudWatch Agent angewiesen, die spezifischen Komponentenmetriken zu sammeln, die in Ihrer Klassifizierungsdatei definiert sind. Die Metriken werden in den in Ihrer Konfiguration angegebenen Intervallen erfasst und veröffentlicht CloudWatch, wo sie visualisiert und analysiert werden können.

Benutzerdefinierte Metriken bieten tiefere Einblicke in die Leistungsmerkmale Ihres Clusters und ermöglichen so eine genauere Überwachung und Fehlerbehebung Ihrer EMR-Anwendungen.

▼ Software settings [Info](#)

Override the default configurations for specific applications on your cluster.

☒ Enter configuration

☐ Load JSON from Amazon S3

```

1  {
2    "Classification": "emr-metrics",
3    "Configurations": [
4      {
5        "Classification": "emr-hadoop-hdfs-datanode-metrics",
6        "Properties": {
7          "Hadoop:service=DataNode,name=DataNodeActivity-*":
8            "DatanodeNetworkErrors,TotalReadTime,TotalWriteTime,BytesRead
9            ,BytesWritten,RemoteBytesRead,RemoteBytesWritten
10           ,ReadBlockOpNumOps,ReadBlockOpAvgTime,WriteBlockOpNumOps
11           ,WriteBlockOpAvgTime",
12          "otel.metric.export.interval": "30000"
13        }
14      },
15      {
16        "Classification": "emr-hadoop-yarn-nodemanager-metrics",
17        "Properties": {
18          "Hadoop:service=NodeManager,name=JvmMetrics": "MemNonHeapUsedM
19          ,MemNonHeapCommittedM,MemNonHeapMaxM,MemHeapUsedM
20          ,MemHeapCommittedM,MemHeapMaxM,MemMaxM",
21          "Hadoop:service=NodeManager,name=NodeManagerMetrics":
22            "ContainerCpuUtilization,NodeCpuUtilization,ContainersCompleted
23            ,ContainersFailed,ContainersKilled,ContainersLaunched
24            ,ContainersRolledBackOnFailure,ContainersRunning
25            ,ContainerUsedMemGB,ContainerUsedVMemGB
26            ,ontainerLaunchDurationNumOps,ContainerLaunchDurationAvgTime",
27          "otel.metric.export.interval": "20000"
28        }
29      }
30    ]
31  }

```

Überschreiben Sie die Standardkonfigurationen.

Aktualisierung der Metrikkonfiguration für laufende Cluster

Sie können die Einstellungen für die Erfassung von Metriken für einen vorhandenen EMR-Cluster ändern, ohne den Betrieb zu unterbrechen, indem Sie die folgenden Schritte ausführen:

1. Navigieren Sie in der AWS Management Console zu Ihrem aktiven EMR-Cluster.
2. Wählen Sie in der Cluster-Detailansicht die Registerkarte Konfigurationen aus.
3. Suchen Sie den Abschnitt Instanzgruppenkonfigurationen.

4. Klicken Sie auf die Schaltfläche „Neu konfigurieren“, um die Einstellungen zu ändern.
5. Wählen Sie „JSON aus Amazon S3 laden“ oder bearbeiten Sie die Konfiguration direkt.
6. Geben Sie den Speicherort Ihrer aktualisierten Metrik-Klassifizierungsdatei ein oder nehmen Sie Änderungen im Editor vor.
7. Wenden Sie die Änderungen an, um das Verhalten bei der Erfassung von Metriken zu aktualisieren.

Diese Rekonfigurationsfunktion ermöglicht es Ihnen, Ihren Überwachungsansatz an die sich ändernden Workload-Anforderungen anzupassen. Der CloudWatch Agent passt sich automatisch an die neue Konfiguration an und erfasst die aktualisierten Messwerte, ohne dass Cluster-Neustarts oder Ausfallzeiten erforderlich sind.

Important

Es kann mehrere Minuten dauern, bis Konfigurationsänderungen auf alle Knoten im Cluster übertragen werden. Überwachen Sie Ihre CloudWatch Dashboards weiterhin, um sicherzustellen, dass die neuen Metriken erwartungsgemäß angezeigt werden.

Properties | Bootstrap actions | Instances (Hardware) | Steps | Applications | **Configurations** | Monitoring | Events | Tags (1)

Cluster configurations View JSON

Cluster configurations are defined when you create a cluster.

Find cluster configurations

Filter classification: Any classification

Classification	Property	Value	Source
spark-hive-site	hive.metastore.client.factory.class	com.amazonaws.glue.catalog.metastore.AWSGlueData...	Cluster configurations
hive-site	hive.metastore.client.factory.class	com.amazonaws.glue.catalog.metastore.AWSGlueData...	Cluster configurations
emr-metrics.emr-hadoop-yarn-resource-manager-metrics	Hadoop:service=ResourceManager,name=CapacitySche...	AllocateNumOps,NodeUpdateNumOps	Cluster configurations
emr-metrics.emr-hadoop-hdfs-namenode-metrics	Hadoop:service=NameNode,name=FSNamesystem	BlockCapacity,CapacityUsedGB	Cluster configurations
emr-metrics.emr-hadoop-hdfs-namenode-metrics	otel.metric.export.interval	20000	Cluster configurations
emr-metrics.emr-hadoop-hdfs-datanode-metrics	Hadoop:service=DataNode,name=JvmMetrics	MemNonHeapUsedM	Cluster configurations
emr-metrics.emr-hadoop-hdfs-datanode-metrics	otel.metric.export.interval	30000	Cluster configurations

Instance group configurations (24) Reconfigure View JSON

Instance groups inherit cluster configurations that are specified when a cluster is created. With EMR 5.21.0 and later, you can override cluster configurations and specify additional configuration classifications for each instance group below as long as its status is **Running**.

Find instance group configurations

Filter instance group: Any instance group | Filter classification: Any classification

Classification	Status	Property	Value	Source
☒ Instance group Task - 1 (ig-1CC61C746LNO9)	Running	-	-	-
☐ Instance group Primary (ig-3INY1F4NK87XL)	Running	-	-	-
☐ Instance group Core (ig-2OH0Y67V0BKTT)	Running	-	-	-

Konfigurationen von Instanzgruppen.

Ihre CloudWatch Integration wird validiert

Nach Abschluss der Konfigurationsschritte ist es an der Zeit, zu überprüfen, ob Ihr Monitoring-Setup ordnungsgemäß funktioniert:

Schritt 1: Bereitstellen Ihres EMR-Clusters

1. Überprüfen Sie alle Konfigurationseinstellungen auf Richtigkeit.
2. Stellen Sie sicher, dass Bootstrap-Aktionen und Klassifizierungsdateien korrekt referenziert werden.
3. Klicken Sie auf Cluster erstellen, um Ihre EMR-Umgebung zu starten.
4. Warten Sie, bis der Cluster den Status Running erreicht hat (normalerweise 5—15 Minuten).

Schritt 2: Testanwendungen ausführen

Reichen Sie mehrere Spark-Testanwendungen ein, um aussagekräftige Metriken zu generieren:

- Führen Sie einen einfachen Spark-Job aus, der Beispieldaten verarbeitet.
- Führen Sie eine länger andauernde Analyseaufgabe aus, um die Ressourcennutzung zu beobachten.
- Testen Sie verschiedene Anwendungskonfigurationen, um Leistungskennzahlen zu vergleichen.

Nachdem Ihre Anwendungen abgeschlossen sind (oder während sie ausgeführt werden):

- Navigieren Sie zur CloudWatch Konsole.
- Suchen Sie in Ihren konfigurierten Protokollgruppen nach Anwendungsprotokollen.
- Untersuchen Sie die Metrik-Dashboards, um CPU-, Speicher- und anwendungsspezifische Metriken zu beobachten.
- Vergewissern Sie sich, dass die in Ihrer Klassifizierungsdatei definierten benutzerdefinierten Metriken in angezeigt werden. CloudWatch

Dieser Validierungsprozess bestätigt, dass Ihre CloudWatch Integration sowohl Protokolle als auch Metriken ordnungsgemäß erfasst, sodass Sie einen umfassenden Einblick in die Leistung und das Anwendungsverhalten Ihres EMR-Clusters erhalten.

Zugreifen auf EMR-Protokolle in CloudWatch Protokollgruppen

Nachdem Ihr EMR-Cluster ausgeführt und der CloudWatch Agent ordnungsgemäß konfiguriert wurde, sind Ihre Anwendungs- und CloudWatch Systemprotokolle in Logs verfügbar. Gehen Sie wie folgt vor, um auf sie zuzugreifen und sie zu analysieren:

Ihre Protokollgruppen anzeigen

1. Navigieren Sie in der AWS Management CloudWatch Console zur Konsole.
2. Wählen Sie im linken Navigationsbereich Protokollgruppen aus.
3. Suchen Sie nach den Protokollgruppen, die durch Ihre Konfiguration erstellt wurden, z. B.:
 - `/emr/yarn/resourcemngernach ResourceManager YARN-Protokollen`.
 - `/emr/hdfs/namenodefür NameNode HDFS-Protokolle`.
 - Alle zusätzlichen Protokollgruppen, die in Ihrer Konfigurationsdatei angegeben sind.

Jede Protokollgruppe enthält Protokollstreams, die nach Instanz-ID organisiert sind, sodass Sie Protokolle zu bestimmten Knoten in Ihrem Cluster zurückverfolgen können.

Mit Protokolldaten arbeiten

- **Protokolldaten durchsuchen:** Verwenden Sie CloudWatch Logs Insights, um strukturierte Abfragen in Ihren Loggruppen durchzuführen.
- **Metriken erstellen:** Extrahieren Sie Metriken aus Protokollmustern, um benutzerdefinierte CloudWatch Metriken zu erstellen.
- **Benachrichtigungen einrichten:** Konfigurieren Sie Alarme auf der Grundlage bestimmter Fehlermuster oder Protokollfrequenzen.
- **Protokolle exportieren:** Laden Sie Protokolle für die Offline-Analyse oder Archivierung herunter.

Aufbewahrung von Protokollen

Note

Standardmäßig werden Protokolle 30 Tage lang aufbewahrt. Sie können die Aufbewahrungsrichtlinie für jede Protokollgruppe ändern, um Protokolle für längere Zeiträume aufzubewahren, falls dies für Compliance- oder Analysezwecke erforderlich ist.

CloudWatch Logs bietet einen zentralen Speicherort für all Ihre EMR-Protokolldaten, sodass Sie keine SSH-Verbindung zu einzelnen Clusterknoten herstellen müssen, um Probleme zu beheben oder das Anwendungsverhalten zu analysieren.

Benutzerdefinierte Metriken im EMR Monitoring Dashboard anzeigen

Nachdem Ihr EMR-Cluster mit der Konfiguration des CloudWatch Agenten und der benutzerdefinierten Metriken ausgeführt wurde, können Sie diese Metriken einfach direkt in der EMR-Konsole überwachen:

Zugriff auf Ihre benutzerdefinierten Metriken

1. Navigieren Sie in der AWS Management Console zu Ihrem EMR-Cluster.
2. Wählen Sie auf der Seite mit den Cluster-Details die Registerkarte Überwachung aus.
3. Suchen Sie oben in den Monitoring-Dashboards nach der Dropdownliste nach der Klassifizierung von Filtermetriken.
4. Verwenden Sie diesen Filter, um bestimmte Metrikkategorien auszuwählen:
 - Wählen Sie HDFS für die Anzeige NameNode und die DataNode Metriken aus.
 - Wählen Sie YARN aus, um Metriken zu sehen ResourceManager und zu speichern.
 - Wählen Sie HBase für HBase -spezifische Leistungsdaten.
 - Wählen Sie benutzerdefinierte metrische Klassifizierungen aus, die Sie definiert haben.

Das Dashboard wird dynamisch aktualisiert und zeigt Diagramme für Ihre ausgewählten Kennzahlen an, die Leistungstrends im Zeitverlauf zeigen.

Arbeiten mit metrischen Visualisierungen

- Zeitbereiche anpassen: Ändern Sie das Zeitfenster, um aktuelle Aktivitäten oder historische Trends anzuzeigen.
- Metriken vergleichen: Zeigen Sie mehrere verwandte Metriken side-by-side für die Korrelationsanalyse an.
- Zoom-Funktionen: Konzentrieren Sie sich auf bestimmte Zeiträume, in denen Anomalien oder Muster auftreten.
- Daten aktualisieren: Aktualisieren Sie Visualisierungen nahezu in Echtzeit mit den neuesten Metrikdaten.

Dieser integrierte Überwachungsansatz ermöglicht es Ihnen, sowohl standardmäßige EMR-Metriken als auch Ihre benutzerdefinierten Messwerte in einem einheitlichen Dashboard zu verfolgen, sodass Sie Leistungsprobleme, Ressourcenbeschränkungen oder Anwendungsengpässe leichter identifizieren können, ohne die EMR-Konsole verlassen zu müssen.

The screenshot displays the Amazon EMR console's 'Monitoring' tab for a cluster named 'TestCluster'. The 'Monitoring' tab is highlighted with a red box. Below the navigation bar, the 'Amazon CloudWatch metrics' section is visible, with 'CloudWatch agent - new' selected. The 'Filter dashboard' dropdown is set to 'Primary (MASTER - ig-3INY1F4NK87XL)'. A 'Filter metric classification' dropdown menu is open, showing a list of metrics: 'emr-system-metrics' (selected with a checkmark), 'emr-hadoop-hdfs-datanode-metrics', 'emr-hadoop-hdfs-namenode-metrics', and 'emr-hadoop-yarn-resourcemanager-metrics'. The main area shows four placeholder charts for metrics like 'system.cpu_usage_active', 'system.cpu_usage', 'system.cpu_usage_idle', and 'system.cpu_usage_idle'. Each chart displays 'Nodes Reporting (%)' on the y-axis (0 to 100.00) and a time range from 15:00 to 17:00 on the x-axis. The charts indicate 'No data available. Try adjusting the dashboard time range.'

Klassifizierung von Metriken filtern.

Problembehandlung bei Amazon EMR-Clustern

Ein EMR-Cluster läuft in einem komplexen Ökosystem, das Open-Source-Software, benutzerdefinierten Anwendungscode und umfasst AWS-Services. Wenn bei einem dieser Teile ein Problem auftritt, schlägt der Cluster möglicherweise fehl oder es dauert länger als erwartet, bis er abgeschlossen ist. Die folgenden Themen können Ihnen bei der Identifizierung von Cluster-Problemen und deren Behebung helfen.

Themen

- [Welche Tools sind für die Fehlerbehebung bei einem Amazon EMR-Cluster verfügbar?](#)
- [Anzeigen und Neustarten von Amazon-EMR- und Anwendungsprozessen \(Daemons\)](#)
- [Sammlungen häufiger Fehler in Amazon EMR](#)
- [Fehlerbehebung bei einem Amazon EMR-Cluster, der mit einem Fehlercode ausgefallen ist](#)
- [Fehlerbehebung bei einem langsamen Amazon EMR-Cluster](#)
- [Behebung häufiger Probleme bei der Verwendung von Amazon EMR mit AWS Lake Formation](#)

Wenn Sie eine neue Hadoop-Anwendung entwickeln, empfehlen wir Ihnen, das Debugging zu aktivieren und eine kleine, aber repräsentative Teilmenge Ihrer Daten zu verarbeiten, um die Anwendung zu testen. Möglicherweise möchten Sie die Anwendung auch ausführen, um jeden Schritt separat step-by-step zu testen. Weitere Informationen erhalten Sie unter [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#) und [Schritt 5: Testen Sie den Amazon EMR-Cluster Schritt für Schritt](#).

Welche Tools sind für die Fehlerbehebung bei einem Amazon EMR-Cluster verfügbar?

Um Clusterfehler zu identifizieren und zu beheben, können Sie die auf dieser Seite beschriebenen Tools verwenden. Möglicherweise müssen Sie einige der Tools initialisieren, wenn Sie den Cluster starten. Andere Tools sind standardmäßig für jeden Cluster verfügbar.

Themen

- [Anzeigen von EMR Cluster-Details](#)
- [EMR-Cluster-Fehlerdetails anzeigen](#)
- [Skripts ausführen und Amazon-EMR-Prozesse konfigurieren](#)

- [Anzeige von -Protokolldateien](#)
- [Überwachen Sie die Leistung des EMR-Clusters](#)

Anzeigen von EMR Cluster-Details

Sie können die AWS Management Console, oder EMR-API verwenden AWS CLI, um detaillierte Informationen über einen EMR-Cluster und die Jobausführung abzurufen. Weitere Hinweise zur Verwendung von AWS Management Console und finden Sie AWS CLI unter. [Status und Details des Amazon EMR-Clusters anzeigen](#)

Detailbereich der Amazon-EMR-Konsole

In der Liste Cluster in der Amazon-EMR-Konsole werden allgemeine Informationen über den Status der einzelnen Cluster in Ihrem Konto und Ihrer AWS-Region angezeigt. Die Liste zeigt alle aktiven und beendeten Cluster an, die Sie in den vergangenen zwei Monaten gestartet haben. Sie können in der Liste Clusters (Cluster) den Name (Namen) eines Clusters auswählen, um Details zu diesem anzuzeigen. Diese Informationen sind in verschiedene Kategorien unterteilt, um das Navigieren zu vereinfachen.

Die auf der Cluster-Detailseite verfügbaren Anwendungsbenutzeroberflächen können bei der Fehlerbehebung bei Clustern hilfreich sein. Sie zeigt den Status von YARN-Anwendungen. Bei einigen Anwendungen wie z. B. Spark-Anwendungen können Sie verschiedene Metriken und Facets wie Aufträge, Phasen und Ausführende anzeigen. Weitere Informationen finden Sie unter [Amazon EMR-Anwendungsverlauf anzeigen](#). Dieses Feature ist nur mit Amazon-EMR-Versionen 5.8.0 und höher verfügbar.

Amazon-EMR;-Befehlszeilenschnittstelle

Einzelheiten zu einem Cluster finden Sie in dem AWS CLI mit dem `--describe` Argument.

Amazon-EMR-API

Sie können die Details eines Cluster in der API mit der Aktion `DescribeJobFlows` abrufen.

EMR-Cluster-Fehlerdetails anzeigen

Wenn ein EMR-Cluster mit einem Fehler beendet wird, werden ein Fehlercode und eine Fehlermeldung `ListClusters` APIs zurückgegeben. `DescribeCluster` Bei ausgewählten Clusterfehlern kann Ihnen das `ErrorDetail`-Datenarray bei der Behebung des Fehlers helfen.

Eine Liste der Fehlercodes, die `ErrorDetail` Daten enthalten, finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#).

 Note

Wir verfeinern unsere Fehlermeldungen kontinuierlich, damit Sie die aktuellsten und relevantesten Informationen erhalten. Es wird nicht empfohlen, den Text von `ErrorMessage` zu analysieren, da sich dieser Text ändern kann.

Skripts ausführen und Amazon-EMR-Prozesse konfigurieren

Im Rahmen Ihrer Problembehandlung kann es hilfreich sein, benutzerdefinierte Skripts auf Ihrem Cluster auszuführen oder Clusterprozesse anzuzeigen und zu konfigurieren.

Anwendungsprozesse anzeigen und neu starten

Es kann hilfreich sein, sich die laufenden Prozesse auf Ihrem Cluster anzusehen, um potenzielle Probleme zu diagnostizieren. Sie können Clusterprozesse beenden und neu starten, indem Sie eine Verbindung zum Hauptknoten Ihres Clusters herstellen. Weitere Informationen finden Sie unter [Anzeigen und Neustarten von Amazon-EMR- und Anwendungsprozessen \(Daemons\)](#).

Führen Sie Befehle und Skripts ohne SSH-Verbindung aus

Um als Schritt einen Befehl oder ein Skript auf Ihrem Cluster auszuführen, können Sie die Tools `command-runner.jar` oder `script-runner.jar` verwenden, ohne eine SSH-Verbindung zum Hauptknoten herzustellen. Weitere Informationen finden Sie unter [Befehle und Skripts auf einem Amazon-EMR-Cluster ausführen](#).

Anzeige von -Protokolldateien

Amazon EMR und Hadoop generieren beide Protokolldateien, während der Cluster ausgeführt wird. Sie können auf diese Protokolldateien mit mehreren Tools zugreifen, abhängig von der Konfiguration, die Sie beim Starten des Clusters angegeben haben. Weitere Informationen finden Sie unter [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#).

Protokolldateien auf dem Hauptknoten

Jeder Cluster veröffentlicht Protokolldateien im Verzeichnis `/mnt/var/log/` auf dem Master-Knoten. Diese Protokolldateien sind nur verfügbar, während der Cluster ausgeführt wird.

So archivieren Sie Protokolldateien in Amazon S3

Wenn Sie den Cluster starten und einen Amazon S3-Protokollpfad angeben, kopiert der Cluster die in `/mnt/var/log/` auf dem Master-Knoten gespeicherten Protokolldateien in 5-Minuten-Intervallen nach Amazon S3. So wird sichergestellt, dass Sie Zugriff auf die Protokolldateien auch nach Beendigung des Clusters haben. Da die Dateien in 5-Minuten-Intervallen archiviert werden, stehen die letzten Minuten eines unvermittelt beendeten Clusters ggf. nicht zur Verfügung.

Überwachen Sie die Leistung des EMR-Clusters

Amazon EMR bietet mehrere Tools zur Überwachung der Leistung Ihres Clusters.

Hadoop-Webschnittstellen

Jeder Cluster veröffentlicht eine Reihe von Webschnittstellen auf dem Master-Knoten, die Informationen über den Cluster enthalten. Sie können auf diese Webseiten über einen SSH-Tunnel zugreifen, um sie auf dem Master-Knoten zu verbinden. Weitere Informationen finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

CloudWatch Metriken

Jeder Cluster meldet Metriken an CloudWatch. CloudWatch ist ein Webservice, der Metriken verfolgt und mit dem Sie Alarme für diese Metriken einrichten können. Weitere Informationen finden Sie unter [Überwachung von Amazon EMR-Metriken mit CloudWatch](#).

Anzeigen und Neustarten von Amazon-EMR- und Anwendungsprozessen (Daemons)

Wenn Sie in einem Cluster Fehler beheben, möchten Sie möglicherweise laufende Prozesse auflisten. Möglicherweise möchten Sie Prozesse auch beenden oder neu starten. Sie können beispielsweise einen Prozess neu starten, nachdem Sie eine Konfiguration geändert haben, oder ein Problem mit einem bestimmten Prozess feststellen, nachdem Sie Protokolldateien und Fehlermeldungen analysiert haben.

Es gibt zwei Arten von Prozessen, die auf einem Cluster ausgeführt werden: Amazon EMR-Prozesse (z. B. Instance-Controller und Log Pusher) und Prozesse, die mit den auf dem Cluster installierten Anwendungen verknüpft sind (z. B., `hadoop-hdfs-namenode` und `hadoop-yarn-resourcemanager`).

Um mit Prozessen direkt auf einem Cluster zu arbeiten, stellen Sie eine Verbindung mit dem Hauptknoten her. Weitere Informationen finden Sie unter [Eine Verbindung zu einem Amazon-EMR-Cluster herstellen](#).

Anzeigen von ausgeführten Prozessen

Die Methode, mit der Sie laufende Prozesse in einem Cluster anzeigen, unterscheidet sich je nach der von Ihnen verwendeten Amazon-EMR-Version.

EMR 5.30.0 and 6.0.0 and later

Example : Listet alle laufenden Prozesse auf

Im folgenden Beispiel wird `systemctl` verwendet und `--type` angegeben, um alle Prozesse anzuzeigen.

```
systemctl --type=service
```

Example : Listet bestimmte Prozesse auf

Im folgenden Beispiel werden alle Prozesse aufgeführt, deren Namen `hadoop` enthalten.

```
systemctl --type=service | grep -i hadoop
```

Beispielausgabe:

<code>hadoop-hdfs-namenode.service</code>	<code>loaded active running Hadoop namenode</code>
<code>hadoop-httpfs.service</code>	<code>loaded active running Hadoop httpfs</code>
<code>hadoop-kms.service</code>	<code>loaded active running Hadoop kms</code>
<code>hadoop-mapreduce-historyserver.service</code>	<code>loaded active running Hadoop historyserver</code>
<code>hadoop-state-pusher.service</code>	<code>loaded active running Daemon process that processes and serves EMR metrics data.</code>
<code>hadoop-yarn-proxyserver.service</code>	<code>loaded active running Hadoop proxyserver</code>
<code>hadoop-yarn-resourcemanager.service</code>	<code>loaded active running Hadoop resourcemanager</code>
<code>hadoop-yarn-timelineserver.service</code>	<code>loaded active running Hadoop timelineserver</code>

Example : Sehen Sie sich einen detaillierten Statusbericht für einen bestimmten Prozess an

Im folgenden Beispiel wird ein detaillierter Statusbericht für den `hadoop-hdfs-namenode-Service` angezeigt.

```
sudo systemctl status hadoop-hdfs-namenode
```

Beispielausgabe:

```
hadoop-hdfs-namenode.service - Hadoop namenode
  Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
  preset: disabled)
  Active: active (running) since Wed 2021-08-18 21:01:46 UTC; 26min ago
  Main PID: 9733 (java)
  Tasks: 0
  Memory: 1.1M
  CGroup: /system.slice/hadoop-hdfs-namenode.service
          # 9733 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server -
  XX:OnOutOfMemoryError=kill -9 %p ...

Aug 18 21:01:37 ip-172-31-20-123 systemd[1]: Starting Hadoop namenode...
Aug 18 21:01:37 ip-172-31-20-123 su[9715]: (to hdfs) root on none
Aug 18 21:01:37 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: starting namenode,
  logging to /var/log/hadoop-hdfs/ha...out
Aug 18 21:01:46 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: Started Hadoop
  namenode: [ OK ]
Aug 18 21:01:46 ip-172-31-20-123 systemd[1]: Started Hadoop namenode.
Hint: Some lines were ellipsized, use -l to show in full.
```

EMR 4.x - 5.29.0

Example : Listet alle laufenden Prozesse auf

Das folgende Beispiel listet alle laufenden Prozesse auf.

```
initctl list
```

EMR 2.x - 3.x

Example : Listet alle laufenden Prozesse auf

Das folgende Beispiel listet alle laufenden Prozesse auf.

```
ls /etc/init.d/
```

Beenden und Neustarten von Prozessen

Nachdem Sie bestimmen, welche Prozesse ausgeführt werden, können Sie diese beenden und dann neu starten.

EMR 5.30.0 and 6.0.0 and later

Example : Stoppt einen Prozess

Das folgende Beispiel stoppt den `hadoop-hdfs-namenode`-Prozess.

```
sudo systemctl stop hadoop-hdfs-namenode
```

Sie können `status` abfragen, um zu überprüfen, ob der Prozess gestoppt wurde.

```
sudo systemctl status hadoop-hdfs-namenode
```

Beispielausgabe:

```
hadoop-hdfs-namenode.service - Hadoop namenode
  Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
  preset: disabled)
  Active: failed (Result: exit-code) since Wed 2021-08-18 21:37:50 UTC; 8s ago
  Main PID: 9733 (code=exited, status=143)
```

Example : Startet einen Prozess

Das folgende Beispiel startet den `hadoop-hdfs-namenode`-Prozess.

```
sudo systemctl start hadoop-hdfs-namenode
```

Sie können den Status überprüfen, um sicherzustellen, dass der Prozess ausgeführt wird.

```
sudo systemctl status hadoop-hdfs-namenode
```

Beispielausgabe:

```
hadoop-hdfs-namenode.service - Hadoop namenode
  Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
  preset: disabled)
  Active: active (running) since Wed 2021-08-18 21:38:24 UTC; 2s ago
```



```
Process: 13748 ExecStart=/etc/init.d/hadoop-hdfs-namenode start (code=exited,
status=0/SUCCESS)
Main PID: 13800 (java)
Tasks: 0
Memory: 1.1M
CGroup: /system.slice/hadoop-hdfs-namenode.service
# 13800 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server
-XX:OnOutOfMemoryError=kill -9 %p...
```

EMR 4.x - 5.29.0

Example : Stoppt einen laufenden Prozess

Im folgenden Beispiel wird der `hadoop-hdfs-namenode`-Service gestoppt.

```
sudo stop hadoop-hdfs-namenode
```

Example : Startet einen gestoppten Prozess neu

Im folgenden Beispiel wird der `hadoop-hdfs-namenode`-Service neu gestartet. Sie müssen den `start`-Befehl verwenden und nicht `restart`.

```
sudo start hadoop-hdfs-namenode
```

Example : Überprüfen des Prozessstatus

Im Folgenden wird der Status für `hadoop-hdfs-namenode` abgerufen. Sie können den `status` Befehl verwenden, um zu überprüfen, ob der Prozess gestoppt oder gestartet wurde.

```
sudo status hadoop-hdfs-namenode
```

EMR 2.x - 3.x

Example : Beenden eines Anwendungsprozesses

Im folgenden Beispiel wird der `hadoop-hdfs-namenode`-Service beendet, der mit der auf dem Cluster installierten Version von Amazon EMR verknüpft ist.

```
sudo /etc/init.d/hadoop-hdfs-namenode stop
```

Example : Startet einen Anwendungsprozess neu

Geben Sie den folgenden Befehl ein, um den Prozess `hadoop-hdfs-namenode` neu zu starten:

```
sudo /etc/init.d/hadoop-hdfs-namenode start
```

Example : Beendet einen Amazon-EMR-Prozesses

Das folgende Beispiel stoppt einen Prozess, wie z. B. instance-controller, der nicht mit der Version von Amazon EMR auf dem Cluster verknüpft ist.

```
sudo /sbin/stop instance-controller
```

Example : Neustart eines Amazon-EMR-Prozesses

Im folgenden Beispiel wird ein Prozess neu gestartet, z. B. instance-Controller, der nicht mit der Version von Amazon EMR auf dem Cluster verknüpft ist.

```
sudo /sbin/start instance-controller
```

Note

Die Befehle `/sbin/start`, `stop` und `restart` sind symbolische Links zu `/sbin/initctl`. Weitere Informationen zu `initctl` finden Sie auf der `initctl` man-Seite. Geben Sie `man initctl` in die Befehlszeile ein.

Sammlungen häufiger Fehler in Amazon EMR

Manchmal schlagen Cluster fehl oder verarbeiten Daten nur langsam. In den folgenden Abschnitten werden häufig auftretende Probleme mit Clustern aufgeführt. Zu den Fehlern gehören Bootstrap-Fehler und Validierungsfehler sowie Vorschläge zu deren Behebung.

Themen

- [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#)
- [Ressourcenfehler bei Amazon EMR-Clustervorgängen](#)
- [Cluster-Eingabe- und Ausgabefehler bei Amazon EMR-Vorgängen](#)
- [Berechtigungsfehler bei Amazon EMR-Clustervorgängen](#)
- [Hive-Cluster-Fehler](#)
- [VPC-Fehler bei Amazon EMR-Clusteroperationen](#)

- [Fehler beim Streaming von Amazon EMR-Clustern](#)
- [Amazon EMR: Fehler beim benutzerdefinierten JAR-Cluster](#)
- [Fehler bei Amazon EMR AWS GovCloud \(US-West\)](#)
- [Finden Sie einen fehlenden Cluster](#)

Fehlercodes mit ErrorDetail Informationen in Amazon EMR

Wenn ein EMR-Cluster mit einem Fehler beendet wird, werden ein Fehlercode und eine Fehlermeldung `ListClusters` APIs zurückgegeben. `DescribeCluster` Bei einigen Clusterfehlern kann Ihnen das `ErrorDetail`-Datenarray bei der Behebung des Fehlers helfen.

Fehler, die ein `ErrorDetail`-Array beinhalten, enthalten die folgenden Informationen:

ErrorCode

Ein eindeutiger Fehlercode, den Sie für den programmatischen Zugriff verwenden können.

ErrorData

Eine Liste von Bezeichnern in Schlüssel-Wert-Paaren, die Sie für die Programmierung oder die manuelle Suche verwenden können. Eine Beschreibung der `ErrorData` Werte, die ein Fehlercode enthält, finden Sie auf der Seite zur Problembehandlung für den Fehlercode.

ErrorMessage

Beschreibung des Fehlers mit einem Link zu weiteren Informationen in der Amazon-EMR-Dokumentation.

Note

Es wird nicht empfohlen, den Text von `ErrorMessage` zu analysieren, da sich dieser Text ändern kann.

Fehlercodes nach Kategorie

- [Bootstrap-Fehlercodes in Amazon EMR](#)
- [Interne Fehlercodes für Amazon EMR](#)
- [Fehlercodes für Validierungsfehler in Amazon EMR](#)

Bootstrap-Fehlercodes in Amazon EMR

Die folgenden Abschnitte enthalten Informationen zur Fehlerbehebung bei Bootstrap-Fehlercodes.

Themen

- [BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE](#)
- [BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY](#)
- [BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY](#)

BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE

Übersicht

Wenn ein Cluster mit einem `BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE`-Fehler beendet wird, ist eine Bootstrap-Aktion in der primären Instance fehlgeschlagen. Weitere Informationen zu Bootstrap-Aktionen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#).

Auflösung

Um diesen Fehler zu beheben, überprüfen Sie die im API-Fehler zurückgegebenen Details, ändern Sie Ihr Bootstrap-Aktionsskript und erstellen Sie einen neuen Cluster mit der aktualisierten Bootstrap-Aktion.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

primary-instance-id

Die ID der primären Instance, bei der die Bootstrap-Aktion fehlgeschlagen ist.

bootstrap-action

Die Ordinalzahl für die fehlgeschlagene Bootstrap-Aktion. Ein Skript mit dem `bootstrap-action`-Wert von 1 ist die erste Bootstrap-Aktion, die auf der Instance ausgeführt wird.

return-code

Der Rückgabecode für die fehlgeschlagene Bootstrap-Aktion.

amazon-s3-path

Der Amazon-S3-Speicherort der Bootstrap-Aktion, die fehlgeschlagen ist.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Gehen Sie wie folgt vor, um die Ursache des Bootstrap-Aktionsfehlers zu ermitteln und zu beheben. Starten Sie dann einen neuen Cluster.

1. Überprüfen Sie die Bootstrap-Aktionsprotokolldateien in Amazon S3, um die Hauptursache für den Fehler zu ermitteln. Weitere Informationen zum Anzeigen von Amazon-EMR-Protokollen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).
2. Wenn Sie bei der Erstellung der Instance Cluster-Protokolle aktiviert haben, finden Sie weitere Informationen im stdout-Protokoll. Sie finden das stdout-Protokoll für die Bootstrap-Aktion an diesem Amazon-S3-Speicherort:

```
s3://amzn-s3-demo-bucket/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
```

Weitere Informationen zu Clusterprotokollen finden Sie im Abschnitt [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#).

3. Um festzustellen, ob die Bootstrap-Aktion fehlgeschlagen ist, überprüfen Sie die Ausnahmen in den stdout-Protokollen und den return-code-Wert in ErrorData.
4. Verwenden Sie Ihre Ergebnisse aus dem vorherigen Schritt, um Ihre Bootstrap-Aktion so zu überarbeiten, dass Ausnahmen vermieden werden oder Ausnahmen ordnungsgemäß behandelt werden können, wenn sie auftreten.
5. Starten Sie einen neuen Cluster mit Ihrer aktualisierten Bootstrap-Aktion.

BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY

Übersicht

Ein Cluster wird mit dem BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY-Fehler beendet, wenn die primäre Instance kein Bootstrap-Aktionsskript von dem von Ihnen angegebenen Amazon-S3-Speicherort herunterladen kann. Zu den potentiellen Ursachen zählen auch die Folgenden:

- Die Bootstrap-Aktionsskriptdatei befindet sich nicht am angegebenen Amazon-S3-Speicherort.
- Die Servicerolle für EC2 Amazon-Instances auf dem Cluster (auch EC2 Instance-Profil für Amazon EMR genannt) hat keine Berechtigungen für den Zugriff auf den Amazon S3-Bucket, in dem sich das Bootstrap-Aktionsskript befindet. Weitere Informationen zu Servicerollen finden Sie unter [Servicerolle für EC2 Cluster-Instances \(EC2Instance-Profil\)](#).

Weitere Informationen zu Bootstrap-Aktionen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#).

Auflösung

Um diesen Fehler zu beheben, stellen Sie sicher, dass Ihre primäre Instance über angemessenen Zugriff auf das Bootstrap-Aktionsskript verfügt.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

primary-instance-id

Die ID der primären Instance, bei der die Bootstrap-Aktion fehlgeschlagen ist.

bootstrap-action

Die Ordinalzahl für die fehlgeschlagene Bootstrap-Aktion. Ein Skript mit dem `bootstrap-action`-Wert von 1 ist die erste Bootstrap-Aktion, die auf der Instance ausgeführt wird.

amazon-s3-path

Der Amazon-S3-Speicherort der Bootstrap-Aktion, die fehlgeschlagen ist.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Gehen Sie wie folgt vor, um die Ursache des Bootstrap-Aktionsfehlers zu ermitteln und zu beheben. Starten Sie dann einen neuen Cluster.

Fehlerbehebungsschritte

1. Verwenden Sie den `amazon-s3-path`-Wert aus dem `ErrorData`-Array, um das entsprechende Bootstrap-Aktionsskript in Amazon S3 zu finden.
2. Wenn Sie bei der Erstellung der Instance Cluster-Protokolle aktiviert haben, finden Sie weitere Informationen im `stdout`-Protokoll. Sie finden das `stdout`-Protokoll für die Bootstrap-Aktion an diesem Amazon-S3-Speicherort:

```
s3://amzn-s3-demo-bucket/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-  
actions/Failed_Bootstrap_Action_Number/stdout.gz
```

Weitere Informationen zu Clusterprotokollen finden Sie im Abschnitt [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#).

3. Um festzustellen, ob die Bootstrap-Aktion fehlgeschlagen ist, überprüfen Sie die Ausnahmen in den `stdout`-Protokollen und den `return-code`-Wert in `ErrorData`.
4. Verwenden Sie Ihre Ergebnisse aus dem vorherigen Schritt, um Ihre Bootstrap-Aktion so zu überarbeiten, dass Ausnahmen vermieden werden oder Ausnahmen ordnungsgemäß behandelt werden können, wenn sie auftreten.
5. Starten Sie einen neuen Cluster mit Ihrer aktualisierten Bootstrap-Aktion.

BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY

Übersicht

Der `BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY`-Fehler weist darauf hin, dass die primäre Instance das Bootstrap-Aktionsskript nicht finden kann, das die Instance gerade aus dem angegebenen Amazon-S3-Bucket heruntergeladen hat.

Auflösung

Um diesen Fehler zu beheben, stellen Sie sicher, dass Ihre primäre Instance über angemessenen Zugriff auf das Bootstrap-Aktionsskript verfügt.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

primary-instance-id

Die ID der primären Instance, bei der die Bootstrap-Aktion fehlgeschlagen ist.

bootstrap-action

Die Ordinalzahl für die fehlgeschlagene Bootstrap-Aktion. Ein Skript mit dem `bootstrap-action`-Wert von 1 ist die erste Bootstrap-Aktion, die auf der Instance ausgeführt wird.

amazon-s3-path

Der Amazon-S3-Speicherort der Bootstrap-Aktion, die fehlgeschlagen ist.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Gehen Sie wie folgt vor, um die Ursache des Bootstrap-Aktionsfehlers zu ermitteln und zu beheben. Starten Sie dann einen neuen Cluster.

1. Verwenden Sie den `amazon-s3-path`-Wert aus dem `ErrorData`-Array, um das entsprechende Bootstrap-Aktionsskript in Amazon S3 zu finden.
2. Überprüfen Sie die Bootstrap-Aktionsprotokolldateien in Amazon S3, um die Hauptursache für den Fehler zu ermitteln. Weitere Informationen zum Anzeigen von Amazon-EMR-Protokollen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Note

Wenn Sie die Protokolle für Ihren Cluster nicht aktiviert haben, müssen Sie einen neuen Cluster mit denselben Konfigurationen und Bootstrap-Aktionen erstellen. Informationen dazu, wie Sie sicherstellen können, dass die Clusterprotokolle aktiviert sind, finden Sie unter [Amazon EMR-Cluster-Protokollierung und Debugging konfigurieren](#).

3. Überprüfen Sie das `stdout`-Protokoll auf Ihre Bootstrap-Aktionen und stellen Sie sicher, dass es keine benutzerdefinierten Prozesse gibt, die Dateien im `/emr/instance-controller/lib/bootstrap-actions`-Ordner auf Ihren primären Instances löschen. Sie finden das `stdout`-Protokoll für die Bootstrap-Aktion an diesem Amazon-S3-Speicherort:


```
s3://amzn-s3-demo-bucket/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-  
actions/Failed_Bootstrap_Action_Number/stdout.gz
```

4. Starten Sie einen neuen Cluster mit Ihrer aktualisierten Bootstrap-Aktion.

Interne Fehlercodes für Amazon EMR

Die folgenden Abschnitte enthalten Informationen zur Fehlerbehebung für interne Fehlercodes, einschließlich Codes für unzureichende oder fehlende Kapazität.

Themen

- [EC2INTERNAL_ERROR__INSUFFICIENT_CAPACITY_AZ](#)
- [INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY](#)
- [INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY](#)

EC2INTERNAL_ERROR__INSUFFICIENT_CAPACITY_AZ

Übersicht

Ein Cluster wird mit einem `INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ` Fehler beendet, wenn die ausgewählte Availability Zone nicht über genügend Kapazität verfügt, um Ihre Anfrage vom EC2 Amazon-Instance-Typ zu erfüllen. Die Availability Zone ist von dem von Ihnen für einen Cluster ausgewählten Subnetz abhängig. Weitere Informationen über die Amazon-EMR-Unterstützung erhalten Sie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#).

Auflösung

Um diesen Fehler zu beheben, ändern Sie Ihre Instance-Typ-Konfigurationen und erstellen Sie einen neuen Cluster mit Ihrer aktualisierten Anfrage.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

instance-type

Der Instance-Typ, dessen Kapazität aufgebraucht ist.

availability-zone

Die Availability Zone, in die Ihr Subnetz aufgelöst wird.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Gehen Sie wie folgt vor, um die Ursache des Cluster Konfigurationsfehlers zu ermitteln und zu beheben:

- Für andere Clusterkonfiguration lesen Sie die bewährten Methoden. Weitere Informationen finden Sie unter [Konfiguration von Amazon EMR-Cluster-Instance-Typen und Best Practices für Spot-Instances](#) im Verwaltungshandbuch für Amazon EMR.
- Beheben Sie die Startprobleme und überprüfen Sie Ihre Konfiguration. Weitere Informationen finden Sie unter [Problembehandlung beim Starten von Instances](#) im EC2 Amazon-Benutzerhandbuch.
- Starten Sie einen neuen Cluster mit Ihrer aktualisierten Cluster-Konfiguration.

INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY

Übersicht

Ein Cluster wird mit einem INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY-Fehler beendet, wenn Amazon EMR Ihre Spot Instance-Anfrage für den Primärknoten nicht erfüllen kann, weil Instances nicht zu oder unter Ihrem maximalen Spot-Preis verfügbar sind. Weitere Informationen finden Sie unter [Spot-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Auflösung

Um diesen Fehler zu beheben, geben Sie Instance-Typen für Ihren Cluster an, die innerhalb Ihres Preisziels liegen, oder erhöhen Sie Ihr Preislimit für denselben Instance-Typ.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

primary-instance-id

Die ID für die primäre Instance des Clusters, die fehlgeschlagen ist.

instance-type

Der Instance-Typ, dessen Kapazität aufgebraucht ist.

availability-zone

Die Availability Zone, in der sich Ihr Subnetz befindet.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Führen Sie die folgenden Schritte aus, um Probleme mit Ihrer Cluster-Konfigurationsstrategie zu beheben, und starten Sie dann einen neuen Cluster:

1. Lesen Sie die Best Practices für Amazon EC2 Spot-Instances und überprüfen Sie Ihre Cluster-Konfigurationsstrategie. Weitere Informationen finden Sie unter [Bewährte Methoden für EC2 Spot](#) im EC2 Amazon-Benutzerhandbuch und [Konfiguration von Amazon EMR-Cluster-Instance-Typen und Best Practices für Spot-Instances](#).
2. Um diesen Fehler zu beheben, ändern Sie Ihre Instance-Typ-Konfigurationen oder Availability Zone und erstellen Sie einen neuen Cluster mit Ihrer aktualisierten Anfrage.
3. Wenn das Problem weiterhin besteht, verwenden Sie On-Demand-Kapazität für Ihre primäre Instance.

INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY**Übersicht**

Ein Cluster wird mit einem INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY-Fehler beendet, wenn nicht genügend Kapazität vorhanden ist, um eine Spot-Instance-Anfrage für Ihren Primärknoten zu erfüllen. Weitere Informationen finden Sie unter [Spot-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Auflösung

Um diesen Fehler zu beheben, geben Sie Instance-Typen für Ihren Cluster an, die innerhalb Ihres Preisziels liegen, oder erhöhen Sie Ihr Preislimit für denselben Instance-Typ.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

primary-instance-id

Die ID für die primäre Instance des Clusters, die fehlgeschlagen ist.

instance-type

Der Instance-Typ, dessen Kapazität aufgebraucht ist.

availability-zone

Die Availability Zone, in die Ihr Subnetz aufgelöst wird.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Führen Sie die folgenden Schritte aus, um Probleme mit Ihrer Cluster-Konfigurationsstrategie zu beheben, und starten Sie dann einen neuen Cluster:

1. Lesen Sie die Best Practices für Amazon EC2 Spot-Instances und überprüfen Sie Ihre Cluster-Konfigurationsstrategie. Weitere Informationen finden Sie unter [Bewährte Methoden für EC2 Spot](#) im EC2 Amazon-Benutzerhandbuch und [Konfiguration von Amazon EMR-Cluster-Instance-Typen und Best Practices für Spot-Instances](#).
2. Ändern Sie Ihre Instance-Typ-Konfigurationen und erstellen Sie einen neuen Cluster mit Ihrer aktualisierten Anfrage.
3. Wenn das Problem weiterhin besteht, verwenden Sie On-Demand-Kapazität für Ihre primäre Instance.

Fehlercodes für Validierungsfehler in Amazon EMR

Die folgenden Abschnitte enthalten Informationen zur Fehlerbehebung bei Validierung-Fehlercodes.

Themen

- [VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC](#)
- [VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC](#)
- [VALIDATION_ERROR_INVALID_SSH_KEY_NAME](#)
- [VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED](#)

VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC

Übersicht

Wenn Ihr Cluster und die Subnetze, auf die Sie für Ihren Cluster verweisen, zu verschiedenen virtuellen privaten Clouds (VPCs) gehören, wird der Cluster mit einem Fehler beendet.

`VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC` Sie können Cluster mit Amazon EMR mit der Konfiguration der Instance-Flotten in allen Subnetzen in einer VPC starten. Weitere Informationen über Instance-Flotten finden Sie unter [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#) im Verwaltungshandbuch für Amazon EMR.

Auflösung

Verwenden Sie Subnetze, die zur gleichen VPC gehören wie der Cluster, um diesen Fehler zu beheben.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

vpc

Für jedes Subnetz:VPC-Paar die ID für die VPC, der das Subnetz angehört.

subnet

Für jedes Subnetz:VPC-Paar die ID für das Subnetz.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Führen Sie die folgenden Schritte aus, um den Fehler zu identifizieren und zu beheben:

1. Überprüfen Sie die Subnetze IDs , die im `ErrorData` Array aufgeführt sind, und stellen Sie sicher, dass sie zu der VPC gehören, in der Sie den EMR-Cluster starten möchten.
2. Ändern Sie Ihre Subnetzkonfigurationen. Sie können eine der folgenden Methoden verwenden, um alle verfügbaren öffentlichen und privaten Subnetze in einer VPC zu finden.
 - Navigieren Sie zur Amazon-VPC-Konsole. Wählen Sie Subnetze und listen Sie alle Subnetze auf, die sich in Ihrem Cluster befinden. AWS-Region Um nur öffentliche oder private Subnetze zu finden, wenden Sie den Filter Öffentliche Adresse automatisch zuweisen an. IPv4 Um Subnetze in der von Ihrem Cluster verwendeten VPC zu finden und auszuwählen, verwenden Sie die Option Nach VPC filtern. Weitere Informationen zum Erstellen von Subnetzen finden Sie unter [Erstellen eines Subnetzes](#) im Benutzerhandbuch von Amazon Virtual Private Cloud.
 - Verwenden Sie den AWS CLI , um alle verfügbaren öffentlichen und privaten Subnetze in der VPC zu finden, die Ihr Cluster verwendet. Weitere Informationen finden Sie in der [describe-subnets](#)-API. Informationen zum Erstellen neuer Subnetze in einer VPC finden Sie in der [create-subnet](#)-API.
3. Starten Sie einen neuen Cluster mit Subnetzen aus derselben VPC wie der Cluster.

VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC

Übersicht

Wenn Ihr Cluster und die Sicherheitsgruppen, die Sie Ihrem Cluster zuweisen, zu verschiedenen virtuellen privaten Clouds (VPCs) gehören, wird der Cluster mit einem Fehler beendet.

VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Angaben von Amazon-EMR-verwalteten und zusätzlichen Sicherheitsgruppen](#) und [Steuern Sie den Netzwerkverkehr mit Sicherheitsgruppen für Ihren Amazon EMR-Cluster](#).

Auflösung

Um diesen Fehler zu beheben, verwenden Sie Sicherheitsgruppen, die zu derselben VPC gehören wie der Cluster.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

vpc

Für jedes Security-Group:VPC-Paar die ID für die VPC, zu der die Sicherheitsgruppe gehört.

security-group

Für jedes Security-Group:VPC-Paar die ID für die Sicherheitsgruppe.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Führen Sie die folgenden Schritte aus, um den Fehler zu identifizieren und zu beheben:

1. Überprüfen Sie die Sicherheitsgruppe IDs , die im `ErrorData` Array aufgeführt ist, und stellen Sie sicher, dass sie zu der VPC gehören, auf der Sie den EMR-Cluster starten möchten.
2. Navigieren Sie zur Amazon-VPC-Konsole. Wählen Sie Sicherheitsgruppen aus, um alle Sicherheitsgruppen in der ausgewählten Region aufzulisten. Suchen Sie die Sicherheitsgruppen aus derselben VPC wie Ihr Cluster und ändern Sie dann Ihre Sicherheitsgruppenkonfiguration.
3. Starten Sie einen neuen Cluster mit Sicherheitsgruppen aus derselben VPC wie der Cluster.

VALIDATION_ERROR_INVALID_SSH_KEY_NAME

Übersicht

Ein Cluster wird mit einem `VALIDATION_ERROR_INVALID_SSH_KEY_NAME` Fehler beendet, wenn Sie ein EC2 Amazon-Schlüsselpaar verwenden, das für SSH-Verbindungen zur primären Instance nicht gültig ist. Der Name des Schlüsselpaars ist möglicherweise falsch, oder das key pair ist in der

angeforderten Datei nicht vorhanden AWS-Region. Weitere Informationen zu Schlüsselpaaren finden Sie unter [EC2Amazon-Schlüsselpaare und Linux-Instances](#) im EC2 Amazon-Benutzerhandbuch.

Auflösung

Um diesen Fehler zu beheben, erstellen Sie einen neuen Cluster mit einem gültigen SSH-Schlüsselpaarnamen.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

ssh-key

Der Name des SSH-Schlüsselpaars, den Sie bei der Erstellung des Clusters angegeben haben.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Führen Sie die folgenden Schritte aus, um den Fehler zu identifizieren und zu beheben:

1. Überprüfen Sie Ihre *keypair* PEM-Datei und vergewissern Sie sich, dass sie mit dem Namen des SSH-Schlüssels übereinstimmt, den Sie in der Amazon EMR-Konsole sehen.
2. Navigieren Sie zur EC2 Amazon-Konsole. Stellen Sie sicher, dass der SSH-Schlüsselname, den Sie verwendet haben, in dem AWS-Region, den Ihr Cluster verwendet, verfügbar ist. Sie finden Ihre ID AWS-Region neben Ihrer Konto-ID oben im AWS Management Console.
3. Starten Sie einen neuen Cluster mit einem gültigen SSH-Schlüsselnamen.

VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

Übersicht

Ein Cluster wird mit einem `VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED`-Fehler beendet, wenn die AWS-Region und Availability Zones für Ihren Cluster den angegebenen Instance-Typ für eine oder mehrere Instance-Gruppen nicht unterstützen. Amazon EMR unterstützt

möglicherweise einen Instance-Typ in einer Availability Zone innerhalb einer Region, aber nicht in einer anderen. Die Availability Zone innerhalb der Region ist von dem von Ihnen für einen Cluster ausgewählten Subnetz abhängig. Eine Liste der Instance-Typen und -Regionen, die Amazon EMR unterstützt, finden Sie unter [Unterstützte Instance-Typen mit Amazon EMR](#).

Auflösung

Um diesen Fehler zu beheben, geben Sie Instance-Typen für Ihren Cluster an, die Amazon EMR in der Region und Availability Zone unterstützt, in der Sie den Cluster anfordern.

Informationen zur Behebung des ausgefallenen EMR-Clusters finden Sie in den `ErrorDetail` Informationen, die von `DescribeCluster` und `ListClusters` APIs zurückgegeben wurden. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#). Das in `ErrorDetail` enthaltene `ErrorData`-Array gibt die folgenden Informationen für diesen Fehlercode zurück:

instance-types

Die Liste der nicht unterstützten Instance-Typen.

availability-zones

Die Availability Zone Liste, in die Ihr Subnetz aufgelöst wird.

public-doc

Die öffentliche URL der Dokumentation für den Fehlercode.

Schritte zum Absolvieren

Führen Sie die folgenden Schritte aus, um den Fehler zu identifizieren und zu beheben:

1. Verwenden Sie den AWS CLI , um die verfügbaren Instance-Typen in einer Availability Zone abzurufen. Zu diesem Zweck können Sie den [ec2 describe-instance-type-offerings](#) Befehl verwenden, um verfügbare Instance-Typen nach Standort (AWS-Region oder Availability Zone) zu filtern. Beispielsweise können Sie den folgenden Befehl verwenden, um die Instance-Typen anzuzeigen, die in der angegebenen AZ angeboten werden. *us-east-2a*

```
aws ec2 describe-instance-type-offerings --location-type "availability-  
zone" --filters Name=location,Values=us-east-2a --region us-east-2 --query  
"InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

Weitere Informationen darüber, wie Sie verfügbare Instance-Typen ermitteln können, [finden Sie unter Finden Sie einen EC2 Amazon-Instance-Typ](#).

2. Nachdem Sie die Instance-Typen ermittelt haben, die in derselben Region und Availability Zone wie der Cluster verfügbar sind, wählen Sie eine der folgenden Lösungen, um fortzufahren:
 - a. Erstellen Sie einen neuen Cluster und wählen Sie ein Subnetz für den Cluster aus, das sich in einer Availability Zone befindet, in der der von Ihnen ausgewählte Instance-Typ verfügbar ist und von Amazon EMR unterstützt wird.
 - b. Erstellen Sie einen neuen Cluster in derselben Region und demselben EC2 Amazon-Subnetz wie der ausgefallene Cluster, jedoch mit einem Instance-Typ, der an diesem Standort von Amazon EMR unterstützt wird.

Eine Liste der Instance-Typen und -Regionen, die Amazon EMR unterstützt, finden Sie unter [Unterstützte Instance-Typen mit Amazon EMR](#). Einen Vergleich der Funktionen der Instance-Typen finden Sie unter [EC2 Amazon-Instance-Typen](#).

Ressourcenfehler bei Amazon EMR-Clustervorgängen

Die folgenden Fehler werden häufig durch eingeschränkte Ressourcen im Cluster verursacht. Die Anleitung beschreibt die einzelnen Fehler und bietet Hilfe bei der Fehlerbehebung.

Themen

- [Der Amazon EMR-Cluster endet mit NO_SLAVE_LEFT und den Kernknoten FAILED_BY_MASTER](#)
- [Amazon EMR-Clusterfehler: Block kann nicht repliziert werden, es konnte nur auf Nullknoten repliziert werden.](#)
- [Amazon EMR-Clusterfehler: EC2 KONTINGENT ÜBERSCHRITTEN](#)
- [Amazon EMR-Clusterfehler: Zu viele Abruffehler](#)
- [Amazon EMR-Clusterfehler: Datei konnte nur auf 0 statt auf 1 Knoten repliziert werden](#)
- [Amazon EMR-Clusterfehler: Knoten auf der Deni-List-Liste](#)
- [Drosselungsfehler mit einem Amazon EMR-Cluster](#)
- [Amazon EMR-Clusterfehler: Instanztyp wird nicht unterstützt](#)
- [Amazon EMR-Clusterfehler: EC2 Die Kapazität ist erschöpft](#)
- [Amazon EMR-Clusterfehler: HDFS-Replikationsfaktorfehler](#)
- [Amazon EMR-Clusterfehler: HDFS-Fehler mit unzureichendem Speicherplatz](#)

Der Amazon EMR-Cluster endet mit NO_SLAVE_LEFT und den Kernknoten FAILED_BY_MASTER

Dies passiert in der Regel, da der Beendigungsschutz deaktiviert ist, und alle Core-Knoten überschreiten die Datenträger-Speicherkapazität, die durch einen Schwellenwert für die maximale Auslastung in der `yarn-site`-Konfigurationsklassifizierung angegeben ist, die der `yarn-site.xml`-Datei entspricht. Dieser Wert liegt standardmäßig bei 90 %. Wenn die Festplattenauslastung für einen Kernknoten den Auslastungsschwellenwert überschreitet, meldet der YARN-Health Service den Knoten als `NodeManager UNHEALTHY`. Während sich der Knoten in diesem Zustand befindet, wird er von Amazon EMR gesperrt und ihm werden keine YARN-Container zugeordnet. Wenn der Knoten 45 Minuten lang fehlerhaft bleibt, markiert Amazon EMR die zugehörige EC2 Amazon-Instance zur Kündigung als `FAILED_BY_MASTER`. Wenn alle EC2 Amazon-Instances, die mit Kernknoten verknüpft sind, für die Kündigung markiert sind, wird der Cluster mit dem Status beendet, `NO_SLAVE_LEFT` da keine Ressourcen zur Ausführung von Jobs vorhanden sind.

Das Überschreiten der Datenträgenutzung auf einem Core-Knoten könnte eine Kettenreaktion auslösen. Wenn ein einzelner Knoten den Schwellenwert für die Datenträgenutzung aufgrund von HDFS überschreitet, liegen andere Knoten wahrscheinlich auch in der Nähe des Schwellenwerts. Der erste Knoten überschreitet den Schwellenwert für die Datenträgenutzung, daher wird er von Amazon EMR zu einer Sperrliste hinzugefügt. Dies erhöht den Aufwand der Datenträgenutzung für die verbleibenden Knoten, da sie jetzt untereinander HDFS-Daten replizieren, die auf dem gesperrten Knoten verloren gingen. Jeder Knoten wird anschließend auf die gleiche Weise in den Zustand `UNHEALTHY` versetzt und der Cluster wird schließlich beendet.

Bewährte Methoden und Empfehlungen

Konfigurieren von Cluster-Hardware mit ausreichend Speicher

Wenn Sie einen Cluster erstellen, stellen Sie sicher, dass genügend Core-Knoten vorhanden sind und alle über ausreichend Instance-Speicher und EBS-Speicher-Volumes für HDFS verfügen. Weitere Informationen finden Sie unter [Berechnen der erforderlichen HDFS-Kapazität eines Clusters](#). Sie können auch Core-Instances manuell oder mithilfe der automatischen Skalierung zu vorhandenen Instance-Gruppen hinzuzufügen. Die neuen Instances haben dieselbe Speicherkonfiguration wie andere Instances in der Instance-Gruppe. Weitere Informationen finden Sie unter [Verwenden Sie die Amazon EMR-Cluster-Skalierung, um sich an wechselnde Workloads anzupassen](#).

Aktivieren des Beendigungsschutzes

Beendigungsschutz aktivieren. Auf diese Weise können Sie, wenn ein Core-Node auf der Deny-Liste steht, über SSH eine Verbindung zur zugehörigen EC2 Amazon-Instance herstellen, um Fehler zu beheben und Daten wiederherzustellen. Wenn Sie den Kündigungsschutz aktivieren, beachten Sie, dass Amazon EMR die EC2 Amazon-Instance nicht durch eine neue Instance ersetzt. Weitere Informationen finden Sie unter [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).

Erstellen Sie einen Alarm für die MRUnhealthy CloudWatch Nodes-Metrik

Diese Metrik meldet die Anzahl der Knoten mit dem Status UNHEALTHY. Sie entspricht der YARN-Metrik `mapred.resourcemanager.NoOfUnhealthyNodes`. Sie können eine Benachrichtigung für diesen Alarm einrichten, um über fehlerhafte Knoten informiert zu werden, bevor der 45-Minuten-Timeout erreicht ist. Weitere Informationen finden Sie unter [Überwachung von Amazon EMR-Metriken mit CloudWatch](#).

Anpassen von Einstellungen mit `yarn-site`

Die folgenden Einstellungen können an Ihre Anwendungsanforderungen angepasst werden. Beispiel: Sie möchten den Schwellenwert für die Datenträgnutzung erhöhen, bei dem ein Knoten UNHEALTHY melden, indem Sie den Wert von `yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage` erhöhen.

Sie können diese Werte festlegen, wenn Sie einen Cluster mithilfe der `yarn-site`-Konfigurationsklassifizierung erstellen. Weitere Informationen finden Sie unter [Konfigurieren von Anwendungen](#) in den Amazon-EMR-Versionshinweisen. Sie können auch über SSH eine Verbindung zu den EC2 Amazon-Instances herstellen, die den Kernknoten zugeordnet sind, und dann die Werte `/etc/hadoop/conf.empty/yarn-site.xml` mithilfe eines Texteditors hinzufügen. Nachdem Sie die Änderung vorgenommen haben, müssen Sie `hadoop-yarn-nodemanager` wie unten gezeigt neu starten.

Important

Wenn Sie den NodeManager Dienst neu starten, werden aktive YARN-Container beendet, es `yarn.nodemanager.recovery.enabled` sei denn, Sie sind bei der Erstellung des Clusters auf die `true` Verwendung der `yarn-site` Konfigurationsklassifizierung eingestellt. Darüber hinaus müssen Sie über die Eigenschaft `yarn.nodemanager.recovery.dir` das Verzeichnis angeben, in dem der Containerstatus gespeichert werden soll.

```
sudo /sbin/stop hadoop-yarn-nodemanager
sudo /sbin/start hadoop-yarn-nodemanager
```

Weitere Informationen zu den aktuellen yarn-site-Eigenschaften und Standardwerten finden Sie unter [YARN-StandardEinstellungen](#) in der Apache Hadoop-Dokumentation.

Eigenschaft	Standardwert	Beschreibung
yarn.nodemanager. disk-health-checker.interval-ms	120000	Die Häufigkeit (in Sekunden), mit der die Datenträger-Zustandsprüfung ausgeführt wird.
yarn.nodemanager. disk-health-checker. min-healthy-disks	0,25	Der Mindestanteil der Anzahl der Festplatten, die fehlerfrei sein müssen, NodeManager damit neue Container gestartet werden können. Dies entspricht sowohl yarn.nodemanager.local-dirs (standardmäßig /mnt/yarn in Amazon EMR) und yarn.nodemanager.log-dirs (standardmäßig /var/log/hadoop-yarn/containers, was symbolisch mit mnt/var/log/hadoop-yarn/containers in Amazon EMR verknüpft ist).
yarn.nodemanager. disk-health-checker.max-disk-utilization-per-disk-percentage	90.0	Der maximale Prozentsatz der zulässigen Speicherplatzauslastung, ab der ein Datenträger als fehlerhaft markiert wird. Die Werte können zwischen 0,0 und 100,0 liegen. Wenn der Wert größer oder gleich 100 ist, wird NodeManager

Eigenschaft	Standardwert	Beschreibung
		er geprüft, ob eine volle Festplatte vorhanden ist. Dies gilt für <code>yarn-node-manager.local-dirs</code> und <code>yarn.nodemanager.log-dirs</code> .
<code>yarn.nodemanager.disk-health-checker.min-free-space-per-disk-mb</code>	0	Der mindestens erforderliche verfügbare Speicherplatz, damit ein Datenträger verwendet werden kann. Dies gilt für <code>yarn-node-manager.local-dirs</code> und <code>yarn.nodemanager.log-dirs</code> .

Amazon EMR-Clusterfehler: Block kann nicht repliziert werden, es konnte nur auf Nullknoten repliziert werden.

Der Fehler „Replizieren von Block nicht möglich, nur Replizieren auf null Knoten möglich“ tritt in der Regel auf, wenn ein Cluster nicht über genügend HDFS-Speicher verfügt. Dieser Fehler tritt auf, wenn Sie mehr Daten in Ihrem Cluster generieren als in HDFS gespeichert werden können. Sie sehen diesen Fehler nur, während der Cluster ausgeführt wird, da der HDFS-Speicherplatz nach dem Beenden des Auftrags freigegeben wird.

Die Menge an HDFS-Speicherplatz, der einem Cluster zur Verfügung steht, hängt von der Anzahl und dem Typ der EC2 Amazon-Instances ab, die als Kernknoten verwendet werden. Für HDFS-Speicher werden keine Aufgabenknoten verwendet. Der gesamte Festplattenspeicher auf jeder EC2 Amazon-Instance, einschließlich der angehängten EBS-Speichervolumes, steht HDFS zur Verfügung. Weitere Informationen zur Größe des lokalen Speichers für jeden EC2 Instance-Typ finden Sie unter [Instance-Typen und -Familien](#) im EC2 Amazon-Benutzerhandbuch.

Der zweite Faktor, der sich auf die Menge des verfügbaren HDFS Speicherplatzes auswirkt, ist der Replikationsfaktor. Dieser beschreibt die Anzahl von Kopien jedes Datenblocks, die in HDFS zu Redundanz Zwecken gespeichert werden können. Der Replikationsfaktor steigt mit der Anzahl der

Knoten im Cluster: Es gibt 3 Kopien jedes Datenblocks für einen Cluster mit 10 oder mehr Knoten, 2 Kopien jedes Blocks für einen Cluster mit 4 bis 9 Knoten und 1 Kopie (keine Redundanz) für Cluster mit 3 oder weniger Knoten. Der gesamte verfügbare HDFS-Speicherplatz wird durch den Replikationsfaktor dividiert. In einigen Fällen, z. B. bei Erhöhung der Anzahl von Knoten von 9 auf 10, kann der Anstieg des Replikationsfaktors dazu führen, dass der verfügbare HDFS-Speicherplatz verringert wird.

Beispielsweise kann ein Cluster mit 10 Core-Knoten vom Typ m1.large 2.833 GB Speicherplatz für HDFS zur Verfügung stellen ((10 Knoten X 850 GB pro Knoten)/Replikationsfaktor 3).

Wenn Ihr Cluster den HDFS zur Verfügung stehenden Speicherplatz überschreitet, können Sie Ihrem Cluster weitere Core-Knoten hinzufügen oder die Datenkomprimierung verwenden, um mehr HDFS-Speicherplatz zu erstellen. Wenn es sich bei Ihrem Cluster um einen Cluster handelt, der gestoppt und neu gestartet werden kann, sollten Sie die Verwendung von Kernknoten eines größeren EC2 Amazon-Instance-Typs in Betracht ziehen. Sie können auch den Replikationsfaktor anpassen. Beachten Sie, dass durch Verringern des Replikationsfaktors die Redundanz der HDFS-Daten sowie die Cluster-Funktion zur Wiederherstellung von verlorenen oder beschädigten HDFS-Blöcken beeinträchtigt wird.

Amazon EMR-Clusterfehler: EC2 KONTINGENT ÜBERSCHRITTEN

Wenn Sie die Meldung EC2 QUOTA EXCEEDED erhalten, gibt es möglicherweise mehrere Ursachen. Je nach Konfigurationsunterschieden kann es zwischen 5 und 20 Minuten dauern, bis vorherige Cluster beendet und die entsprechenden Ressourcen wieder freigegeben werden. Wenn Sie beim Versuch, einen Cluster zu starten, die Fehlermeldung EC2 QUOTA EXCEEDED erhalten, kann es daran liegen, dass Ressourcen eines kürzlich beendeten Clusters noch nicht zur Verfügung stehen. Diese Meldung kann auch durch die Größenanpassung einer Instance-Gruppe oder Instance-Flotte an eine Zielgröße, die das aktuelle Instance-Kontingent für das Konto überschreitet, verursacht werden. Dies kann manuell oder automatisch durch Auto Scaling geschehen.

Sie können das Problem u. U. mit den folgenden Optionen beheben:

- Folgen Sie den Anweisungen unter [AWS -Service-Quotas](#) in Allgemeine Amazon Web Services-Referenz, um eine Erhöhung des Servicelimits zu beantragen. Für einige APIs ist die Einrichtung einer CloudWatch Veranstaltung möglicherweise eine bessere Option als die Erhöhung der Grenzwerte. Weitere Details finden Sie unter [Wann sollten EMR-Ereignisse eingerichtet werden in CloudWatch](#).
- Wenn einer oder mehrere der aktiven Cluster nicht ausgelastet sind, skalieren Sie Instance-Gruppen oder reduzieren Sie Zielkapazitäten von Instance-Flotten für aktive Cluster.

- Erstellen Sie Cluster mit weniger EC2 Instanzen oder reduzierter Zielkapazität.

Amazon EMR-Clusterfehler: Zu viele Abruffehler

Die Fehlermeldung „Too many fetch-failures (Zu viele Abruffehler)“ oder „Error reading task output (Fehler beim Lesen der Aufgabenausgabe)“ in Schritt- oder Aufgabenversuchsprotokollen gibt an, dass die auszuführende Aufgabe von der Ausgabe einer anderen Aufgabe abhängt. Dies geschieht häufig, wenn eine Reduce-Aufgabe zur Ausführung in die Warteschlange gestellt wird und die Ausgabe einer oder mehrerer Map-Aufgaben erfordert, die jedoch noch nicht verfügbar ist.

Es gibt mehrere Gründe, warum die Ausgabe noch nicht verfügbar ist:

- Die erforderliche Aufgabe befindet sich noch in Bearbeitung. Dies ist oft eine Map-Aufgabe.
- Die Daten sind möglicherweise aufgrund einer schlechten Netzwerkverbindung nicht verfügbar, wenn sie sich auf einer anderen Instance befinden.
- Wenn HDFS verwendet wird, um die Ausgabe abzurufen, kann ein Problem mit HDFS vorliegen.

Der häufigste Grund ist, dass sich die vorherige Aufgabe noch in Bearbeitung befindet. Dies ist besonders wahrscheinlich, wenn die Fehler beim ersten Ausführen der Reduce-Aufgaben auftreten. Sie können prüfen, ob dies der Fall ist, indem Sie sich das Syslog-Protokoll für den Cluster-Schritt ansehen, der den Fehler zurückgibt. Wenn das Syslog den Fortschritt beider Map- und Reduce-Aufgaben belegt, weist dies darauf hin, dass die Reduce-Phase gestartet wurde und einige Map-Aufgaben noch nicht abgeschlossen sind.

Sehen Sie sich in den Protokollen den Prozentsatz für den Map-Fortschritt an, der auf 100 % ansteigt und dann wieder auf einen niedrigeren Wert zurückfällt. Wenn der Map-Prozentsatz 100 % beträgt, bedeutet das nicht, dass alle Map-Aufgaben abgeschlossen sind. Es bedeutet lediglich, dass Hadoop alle Map-Aufgaben ausführt. Wenn dieser Wert unter 100 % fällt, bedeutet dies, dass eine Map-Aufgabe fehlgeschlagen ist und Hadoop je nach Konfiguration versucht, die Aufgabe neu zu planen. Bleibt der Zuordnungsprozentsatz in den Protokollen bei 100%, schauen Sie sich vor allem die CloudWatch Metriken an, um zu überprüfen `RunningMapTasks`, ob die Zuordnungsaufgabe noch bearbeitet wird. Sie finden diese Informationen auch mithilfe der Hadoop-Weboberfläche auf dem Master-Knoten.

Wenn dieses Problem auftritt, können Sie verschiedene Schritte versuchen:

- Weisen Sie die Reduce-Phase an, länger zu warten, bis sie startet. Ändern Sie dazu die Konfigurationseinstellung `mapred.reduce.slowstart.completed.maps` in Hadoop und legen Sie sie

auf einen längeren Zeitraum fest. Weitere Informationen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#).

- Passen Sie die Reducer-Anzahl der gesamten Reducer-Kapazität des Clusters an. Ändern Sie dazu die Konfigurationseinstellung `mapred.reduce.tasks` für den Auftrag in Hadoop.
- Verwenden Sie einen Kombinationsklassencode zum Minimieren der Anzahl der Ausgaben, die abgerufen werden müssen.
- Vergewissern Sie sich, dass es keine Probleme mit dem EC2 Amazon-Service gibt, die sich auf die Netzwerkleistung des Clusters auswirken. Verwenden Sie dazu das [Dashboard zum Servicestatus](#).
- Überprüfen Sie die CPU- und Arbeitsspeicherressourcen für die Instances in Ihrem Cluster, um sicherzustellen, dass Ihre Datenverarbeitung die Ressourcen Ihrer Knoten nicht überlastet. Weitere Informationen finden Sie unter [Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren](#).
- Prüfen Sie die Version des Amazon Machine Image (AMI), das in Ihrem Amazon-EMR-Cluster verwendet wird. Wenn die Version 2.3.0 bis einschließlich 2.4.4 ist, aktualisieren Sie auf eine neuere Version. AMI-Versionen im angegebenen Bereich verwenden eine Jetty-Version, die ggf. keine Ausgabe aus der Map-Phase liefert. Der Abruf-Fehler tritt auf, wenn die Reducer keine Ausgabe aus der Map-Phase abrufen können.

Jetty ist ein Open-Source-HTTP-Server, der für die Maschine-zu-Maschine-Kommunikation innerhalb eines Hadoop-Clusters verwendet wird.

Amazon EMR-Clusterfehler: Datei konnte nur auf 0 statt auf 1 Knoten repliziert werden

Wenn eine Datei in HDFS geschrieben wird, wird sie in mehreren Core-Knoten repliziert. Wenn Sie diesen Fehler sehen, bedeutet dies, dass der NameNode Daemon keine verfügbaren DataNode Instances hat, in die Daten in HDFS geschrieben werden können. Mit anderen Worten, es findet keine Block-Replikation statt. Dieser Fehler kann durch eine Reihe von Problemen verursacht werden:

- Das HDFS-Dateisystem hat keinen verfügbaren Speicherplatz. Dies ist die wahrscheinlichste Ursache.
- DataNode Instanzen waren möglicherweise nicht verfügbar, als der Job ausgeführt wurde.
- DataNode Instanzen wurden möglicherweise für die Kommunikation mit dem Master-Knoten gesperrt.
- Instances in der Core-Instance-Gruppe sind möglicherweise nicht verfügbar.

- Berechtigungen können fehlen. Beispielsweise ist der JobTracker Daemon möglicherweise nicht berechtigt, Job-Tracker-Informationen zu erstellen.
- Die Einstellung für den reservierten Speicherplatz für eine DataNode Instanz ist möglicherweise unzureichend. Stellen Sie fest, ob dies der Fall ist, indem Sie die Konfigurationseinstellung `dfs.datanode.du.reserved` prüfen.

Um zu überprüfen, ob dieses Problem dadurch verursacht wird, dass HDFS nicht mehr genügend Festplattenspeicher hat, sehen Sie sich die `HDFSUtilization` Metrik unter an. CloudWatch Wenn dieser Wert zu hoch ist, können Sie zusätzliche Core-Knoten zum Cluster hinzufügen. Wenn Sie einen Cluster haben, von dem Sie glauben, dass der HDFS-Festplattenspeicher knapp wird, können Sie einen Alarm einrichten, der Sie warnt, wenn der Wert von `HDFSUtilization` einen bestimmten Wert überschreitet. CloudWatch Weitere Informationen erhalten Sie unter [Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters](#) und [Überwachung von Amazon EMR-Metriken mit CloudWatch](#).

Falls HDFS nicht zu wenig Speicherplatz zur Verfügung gestellt hat, überprüfen Sie die DataNode Protokolle, NameNode Protokolle und Netzwerkverbindungen auf andere Probleme, die HDFS daran gehindert haben könnten, Daten zu replizieren. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Amazon EMR-Clusterfehler: Knoten auf der Deni-List-Liste

Der NodeManager Daemon ist für den Start und die Verwaltung von Containern auf Kern- und Taskknoten verantwortlich. Die Container werden dem NodeManager Daemon von dem Daemon zugewiesen, der ResourceManager auf dem Master-Knoten läuft. Der ResourceManager überwacht den NodeManager Knoten über einen Heartbeat.

Es gibt eine Reihe von Situationen, in denen der ResourceManager Daemon eine Liste auflistet und sie aus dem Pool der Knoten entfernt NodeManager, die für die Bearbeitung von Aufgaben zur Verfügung stehen:

- Wenn der in den NodeManager letzten 10 Minuten (600.000 Millisekunden) keinen Heartbeat an den ResourceManager Daemon gesendet hat. Dieser Zeitraum kann über die Konfigurationseinstellung `yarn.nm.liveness-monitor.expiry-interval-ms` festgelegt werden. Weitere Informationen zur Änderung von Yarn-Konfigurationseinstellungen finden Sie unter [Anwendungen konfigurieren](#) im Amazon-EMR-Versionshandbuch.
- NodeManager überprüft den Zustand der Festplatten, der durch und bestimmt wird. `yarn.nodemanager.local-dirs` `yarn.nodemanager.log-dirs` Die Prüfungen umfassen

Berechtigungen und freien Speicherplatz (< 90 %). Wenn eine Festplatte die Prüfung nicht besteht, verwendet sie diese bestimmte Festplatte nicht NodeManager mehr, meldet den Knotenstatus aber trotzdem als fehlerfrei. Wenn mehrere Festplatten die Prüfung nicht bestehen, wird der Knoten als fehlerhaft gemeldet ResourceManager und dem Knoten werden keine neuen Container zugewiesen.

Der Anwendungsmaster kann einen NodeManager Knoten auch ablehnen, wenn er mehr als drei fehlgeschlagene Aufgaben hat. Sie können hierfür mithilfe des Konfigurationsparameters `mapreduce.job.maxtaskfailures.per.tracker` einen höheren Wert einstellen. Andere Konfigurationseinstellungen, die Sie ändern können, steuern, wie oft versucht wird, eine Aufgabe auszuführen, bevor ein Fehler gemeldet wird: `mapreduce.map.max.attempts` für Map-Aufgaben und `mapreduce.reduce.maxattempts` für Reduce-Aufgaben. Weitere Informationen zur Änderung von Konfigurationseinstellungen finden Sie unter [Anwendungen konfigurieren](#) im Amazon-EMR-Versionshandbuch.

Drosselungsfehler mit einem Amazon EMR-Cluster

Die Fehler „Throttled from **Amazon EC2** while launch cluster“ und „Failed to provision instances due to throttling from**Amazon EC2**“ treten auf, wenn Amazon EMR eine Anfrage nicht abschließen kann, weil ein anderer Service die Aktivität gedrosselt hat. Amazon EC2 ist die häufigste Quelle für Drosselungsfehler, aber auch andere Dienste können die Ursache für Drosselungsfehler sein. [AWS Servicebeschränkungen](#) gelten für jede Region, um die Leistung zu verbessern. Ein Drosselungsfehler weist darauf hin, dass Sie das Servicelimit für Ihr Konto in dieser Region überschritten haben.

Mögliche Ursachen

Die häufigste Ursache für EC2 Drosselungsfehler bei Amazon ist eine große Anzahl von Cluster-Instances, die gestartet werden, sodass Ihr Service-Limit für EC2 Instances überschritten wird. Cluster-Instances können aus den folgenden Gründen gestartet werden:

- Es werden neue Cluster erstellt.
- Die Clustergröße wird manuell angepasst. Weitere Informationen finden Sie unter [Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters](#).
- Instance-Gruppen in einem Cluster fügen Instances als Ergebnis einer Auto Scaling-Regel hinzu ("Scale-Out" oder horizontales Skalieren). Weitere Informationen finden Sie unter [Grundlegendes zu Auto-Scaling-Regeln](#).

- Instance-Flotten in einem Cluster fügen Instances hinzu, um eine erhöhte Zielkapazität zu erreichen. Weitere Informationen finden Sie unter [Planung und Konfiguration von Instance-Flotten für Ihren Amazon EMR-Cluster](#).

Es ist auch möglich, dass die Häufigkeit oder Art der API-Anfrage an Amazon zu Drosselungsfehlern EC2 führt. Weitere Informationen darüber, wie Amazon API-Anfragen EC2 drosselt, finden Sie unter [Query API request rate](#) in der Amazon EC2 API-Referenz.

Lösungen

Erwägen Sie die folgenden Lösungen:

- Folgen Sie den Anweisungen unter [AWS -Service-Quotas](#) in Allgemeine Amazon Web Services-Referenz, um eine Erhöhung des Servicelimits zu beantragen. Für einige APIs ist die Einrichtung einer CloudWatch Veranstaltung möglicherweise die bessere Option, als die Limits zu erhöhen. Weitere Details finden Sie unter [Wann sollten EMR-Ereignisse eingerichtet werden in CloudWatch](#).
- Wenn Sie Cluster haben, die nach demselben Zeitplan gestartet werden, z. B. zu Beginn der Stunde, sollten Sie gestaffelte Startzeiten in Betracht ziehen.
- Wenn die Nachfragespitzen für Ihre Cluster zu groß angelegt sind und Sie Ihre Instance-Kapazitäten in regelmäßigen Abständen angeben, sollten Sie Ihre Instance mit Auto Scaling nach Bedarf hinzufügen und entfernen. Auf diese Weise werden Instances effizienter genutzt und können je nach Bedarfsprofil zu jedem beliebigen Zeitpunkt für ein Konto weniger Instances angefordert werden. Weitere Informationen finden Sie unter [Verwenden der automatischen Skalierung mit einer benutzerdefinierten Richtlinie für Instanzgruppen in Amazon EMR](#).

Amazon EMR-Clusterfehler: Instanztyp wird nicht unterstützt

Wenn Sie einen Cluster erstellen und dieser mit der Fehlermeldung „Der angeforderte Instance-Typ *InstanceType* wird in der angeforderten Availability Zone nicht unterstützt“ fehlschlägt, bedeutet dies, dass Sie den Cluster erstellt und einen Instance-Typ für eine oder mehrere Instance-Gruppen angegeben haben, der von Amazon EMR in der Region und Availability Zone, in der der Cluster erstellt wurde, nicht unterstützt wird. Amazon EMR unterstützt möglicherweise einen Instance-Typ in einer Availability Zone innerhalb einer Region und nicht in einer anderen. Die Availability Zone innerhalb der Region ist von dem von Ihnen für einen Cluster ausgewählten Subnetz abhängig.

Lösung

Ermitteln Sie die verfügbaren Instance-Typen in einer Availability Zone mithilfe der AWS CLI

- Verwenden Sie den Befehl `ec2 run-instances` mit der Option `--dry-run`. Ersetzen `m5.xlarge` Sie im folgenden Beispiel durch den Instance-Typ, den Sie verwenden möchten, durch das `ami-035be7bafff33b6b6` mit diesem Instance-Typ verknüpfte AMI und `subnet-12ab3c45` durch ein Subnetz in der Availability Zone, die Sie abfragen möchten.

```
aws ec2 run-instances --instance-type m5.xlarge --dry-run --image-id ami-035be7bafff33b6b6 --subnet-id subnet-12ab3c45
```

Anweisungen zum Suchen einer AMI-ID finden Sie unter [Suchen eines Linux-AMI](#). Um eine Subnetz-ID zu finden, können Sie den Befehl [describe-subnets](#) verwenden.

Weitere Informationen darüber, wie Sie verfügbare Instance-Typen ermitteln können, [finden Sie unter Finden Sie einen EC2 Amazon-Instance-Typ](#).

Nachdem Sie die verfügbaren Instance-Typen bestimmt haben, können Sie beliebige der folgenden Aktionen ausführen:

- Erstellen Sie den Cluster in derselben Region und demselben EC2 Subnetz und wählen Sie einen anderen Instance-Typ mit ähnlichen Funktionen wie Sie ursprünglich ausgewählt haben. Eine Liste mit unterstützten Instance-Typen finden Sie unter [Unterstützte Instance-Typen mit Amazon EMR](#). Informationen zum Vergleich der Funktionen von EC2 Instance-Typen finden Sie unter [EC2 Amazon-Instance-Typen](#).
- Wählen Sie ein Subnetz für den Cluster in einer Availability Zone aus, in der der Instance-Typ verfügbar ist und von Amazon EMR unterstützt wird.

Vermeiden Sie Fehler beim Start von Instance-Flotten und Clustern, die auf nicht unterstützte primäre Instance-Typen in Amazon EMR zurückzuführen sind

Primäre Knoten sind in Amazon EMR-Clustern unverzichtbar. Ein EMR-Cluster-Start schlägt möglicherweise mit einem `instance type not supported` Fehler fehl, wenn Amazon EMR versucht, den Cluster in einer Availability Zone zu starten, in der der primäre Instance-Typ nicht unterstützt wird. Die erweiterte Availability Zone-Auswahl für Instance-Flottencluster in Amazon EMR filtert automatisch nicht unterstützte AZs Instance-Typen heraus, die Sie in der Cluster-Konfiguration angegeben haben. Das bedeutet, dass Amazon EMR keine Availability Zone auswählt, in der

die konfigurierten primären Instance-Typen nicht unterstützt werden, wodurch Cluster-Startfehler aufgrund nicht unterstützter Instance-Typen verhindert werden.

Um diese Verbesserung zu aktivieren, fügen Sie der Servicerolle oder -richtlinie für Ihren Cluster die erforderliche Berechtigung hinzu. Die neueste Version von `AmazonEMRServicePolicy_v2` beinhaltet diese Berechtigung. Wenn Sie also diese Richtlinie verwenden, ist die Verbesserung bereits verfügbar. Wenn Sie eine benutzerdefinierte Servicerolle oder Richtlinie verwenden, fügen Sie die Berechtigung hinzu, `ec2:DescribeInstanceTypeOfferings` wenn Sie Ihren Cluster starten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "ec2:DescribeInstanceTypeOfferings",
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon EMR-Clusterfehler: EC2 Die Kapazität ist erschöpft

Wenn Sie versuchen, in einer EC2 Availability Zone, in der es keine weiteren Instanzen des angegebenen EC2 Instance-Typs gibt, einen Cluster zu erstellen oder Instances zu einem Cluster hinzuzufügen, tritt der *InstanceType* Fehler „Nicht genügend Kapazität für“ auf. Die Availability Zone ist von dem von Ihnen für einen Cluster ausgewählten Subnetz abhängig.

Um einen Cluster zu erstellen, führen Sie einen der folgenden Schritte aus:

- Geben Sie einen anderen Instance-Typ mit ähnlichen Funktionen an
- Erstellen des Clusters in einer anderen Region
- Wählen Sie ein Subnetz in einer Availability Zone aus, in dem der gewünschte Instance-Typ möglicherweise verfügbar ist.

Führen Sie einen der folgenden Schritte aus, um Instances zu einem laufenden Cluster hinzuzufügen:

- Ändern Sie Instance-Gruppenkonfigurationen oder Instance-Flottenkonfigurationen so bearbeiten, dass verfügbare Instance-Typen mit ähnlichen Funktionen hinzugefügt werden. Eine Liste mit unterstützten Instance-Typen finden Sie unter [Unterstützte Instance-Typen mit Amazon EMR](#).

Informationen zum Vergleich der Funktionen von EC2 Instance-Typen finden Sie unter [EC2 Amazon-Instance-Typen](#).

- Beenden Sie den Cluster und erstellen Sie ihn in einer Region und Verfügbarkeitszone neu, in der der Instancetyp verfügbar ist.

Amazon EMR-Clusterfehler: HDFS-Replikationsfaktorfehler

Wenn Sie einen Core-Knoten aus einer [Core-Instance-Gruppe oder Instance-Flotte](#) entfernen, tritt bei Amazon EMR möglicherweise ein HDFS-Replikationsfehler auf. Dieser Fehler tritt auf, wenn Sie Kernknoten entfernen und die Anzahl der Kernknoten unter den konfigurierten [dfs.replication-Faktor](#) für das Hadoop Distributed File System (HDFS) fällt. Daher kann Amazon EMR den Vorgang nicht sicher ausführen. Um den Standardwert der Konfiguration zu ermitteln, `dfs.replication` [HDFS-Konfiguration](#).

Mögliche Ursachen

Im Folgenden finden Sie die möglichen Ursachen für einen HDFS-Replikationsfaktorfehler:

- Wenn Sie die [Größe einer Core-Instanzgruppe oder Instanzflotte manuell](#) unter den konfigurierten `dfs.replication` Faktor ändern.
- Ihre Richtlinien für [verwaltete Skalierung](#) oder [Autoscaling](#) ermöglichen möglicherweise eine Skalierung, um die Anzahl der Kernknoten unter den Schwellenwert von zu reduzieren. `dfs.replication`
- Dieser Fehler kann auch auftreten, wenn Amazon EMR versucht, einen fehlerhaften Kernknoten zu [ersetzen](#), obwohl ein Cluster die minimale Anzahl von Kernknoten hat, die von definiert ist. [dfs.replication](#)

Lösungen und bewährte Methoden

Im Folgenden finden Sie Lösungen und bewährte Verfahren:

- Wenn Sie die Größe eines Amazon EMR-Clusters manuell ändern, sollten Sie nicht unter den Wert herunterskalieren, `dfs.replication` da Amazon EMR die Größenänderung nicht sicher abschließen kann.
- Wenn Sie Managed Scaling oder Autoscaling verwenden, stellen Sie sicher, dass die Mindestkapazität Ihres Clusters nicht unter dem Faktor liegt. `dfs.replication`

- Die Anzahl der Core-Instances sollte mindestens `dfs.replication` plus eins sein. Dadurch wird sichergestellt, dass Amazon EMR einen fehlerhaften Core-Knoten erfolgreich ersetzen kann, wenn Sie den Austausch fehlerhafter Kerne aktiviert haben.

Important

Der Ausfall eines einzelnen Core-Knotens kann zu HDFS-Datenverlust führen, wenn Sie auf 1 setzen. `dfs.replication` Wenn Ihr Cluster über HDFS-Speicher verfügt, empfehlen wir, den Cluster mit mindestens vier Kernknoten für Produktionsworkloads zu konfigurieren, um Datenverlust zu vermeiden, und außerdem den `dfs.replication` Faktor auf mindestens 2 festzulegen.

Amazon EMR-Clusterfehler: HDFS-Fehler mit unzureichendem Speicherplatz

Wenn Sie versuchen, einen Core-Knoten zu entfernen, Amazon EMR den Vorgang jedoch nicht sicher abschließen kann, weil im HDFS nicht genügend Speicherplatz vorhanden ist, kann es zu einem Fehler im Hadoop Distributed File System (HDFS) kommen. Bevor Amazon EMR einen Core-Knoten entfernt, müssen alle HDFS-Daten auf dem Knoten auf andere Core-Knoten übertragen werden, um Datenredundanz zu gewährleisten. Wenn auf den anderen Kernknoten jedoch nicht genügend Speicherplatz für die Replikation vorhanden ist, kann Amazon EMR den Knoten nicht ordnungsgemäß außer Betrieb nehmen.

Mögliche Ursachen

Im Folgenden finden Sie eine Liste der möglichen Ursachen für den Fehler, dass HDFS nicht genügend Speicherplatz hat:

- Wenn Sie eine Core-Instanzgruppe oder Instanzflotte manuell herunterskalieren, obwohl auf den verbleibenden Knoten vor dem Herunterskalieren nicht genügend HDFS-Speicherplatz für die Datenreplikation vorhanden ist.
- Durch verwaltetes Skalieren oder Autoscaling wird eine Core-Instanzgruppe oder Instanzflotte herunterskaliert, wenn nicht genügend HDFS-Speicherplatz für die Datenreplikation vorhanden ist.
- Amazon EMR versucht, einen fehlerhaften Core-Knoten zu ersetzen, kann den Knoten jedoch aufgrund des unzureichenden HDFS-Speicherplatzes nicht sicher ersetzen.

Lösungen und bewährte Methoden

Im Folgenden finden Sie Lösungen und bewährte Verfahren:

- Skalieren Sie die Anzahl der Kernknoten in Ihrem Amazon EMR-Cluster. Wenn Sie Managed Scaling oder Autoscaling verwenden, erhöhen Sie die Mindestkapazität Ihrer Kernknoten.
- Verwenden Sie größere EBS-Volumes für Ihre Kernknoten, wenn Sie Ihren EMR-Cluster erstellen.
- Löschen Sie nicht benötigte HDFS-Daten in Ihrem EMR-Cluster. Wir empfehlen Ihnen, CloudWatch Alarmer einzurichten, um die HDFSUtilization Metrik in Ihrem Cluster zu überwachen, um festzustellen, ob Ihr EMR-Cluster wenig Speicherplatz hat.

Cluster-Eingabe- und Ausgabefehler bei Amazon EMR-Vorgängen

Die folgenden Fehler treten in Cluster-Ein- und Ausgabeoperationen häufig auf. Verwenden Sie die Anleitungen in diesem Thema, um Fehler zu beheben und Ihre Konfiguration zu überprüfen.

Themen

- [Hat Ihr Pfad zu Amazon Simple Storage Service \(Amazon S3\) mindestens drei Schrägstriche?](#)
- [Versuchen Sie Eingabeverzeichnis rekursiv zu durchlaufen?](#)
- [Ist Ihr Ausgabeverzeichnis bereits vorhanden?](#)
- [Versuche Sie, eine Ressource mit einer HTTP-URL anzugeben?](#)
- [Verweisen Sie mit einem ungültigen Namensformat auf einen Amazon-S3-Bucket?](#)
- [Haben Sie Probleme beim Laden von Daten in oder aus Amazon S3?](#)

Hat Ihr Pfad zu Amazon Simple Storage Service (Amazon S3) mindestens drei Schrägstriche?

Wenn Sie einen Amazon-S3-Bucket angeben, müssen Sie einen Abschlussschrägstrich am Ende der URL anfügen. Anstatt beispielsweise auf einen Bucket als „s3n: //amzn-s3-demo-bucket1" zu verweisen, sollten Sie „s3n: //amzn-s3-demo-bucket1/" verwenden, da Hadoop Ihren Cluster sonst in den meisten Fällen ausfallen lässt.

Versuchen Sie Eingabeverzeichnis rekursiv zu durchlaufen?

Hadoop durchsucht Eingabeverzeichnis nicht rekursiv nach Dateien. Wenn Sie über eine Verzeichnisstruktur wie beispielsweise /corpus/01/01.txt, /corpus/01/02.txt, /corpus/02/01.txt usw.

verfügen und `/corpus/` als Eingabeparameter für Ihren Cluster angeben, findet Hadoop keine Eingabedateien, da das Verzeichnis `/corpus/` leer ist und Hadoop den Inhalt der Unterverzeichnisse nicht überprüft. Entsprechend überprüft Hadoop die Unterverzeichnisse von Amazon-S3-Buckets nicht rekursiv.

Die Eingabedateien müssen sich direkt in dem Eingabeverzeichnis oder dem Amazon-S3-Bucket, das bzw. den Sie angeben, befinden und nicht in Unterverzeichnissen.

Ist Ihr Ausgabeverzeichnis bereits vorhanden?

Wenn Sie einen Ausgabepfad angeben, der bereits vorhanden ist, schlägt der Hadoop-Cluster in den meisten Fällen fehl. Das bedeutet, dass Sie, wenn Sie einen Cluster ausführen und dann diesen Vorgang mit denselben Parametern wiederholen, der erste Lauf und kein weiterer funktioniert. Nach dem ersten Lauf ist der Ausgabepfad vorhanden, was dazu führt, dass alle nachfolgenden Läufe fehlschlagen.

Versuche Sie, eine Ressource mit einer HTTP-URL anzugeben?

Hadoop akzeptiert keine Ressourcenspeicherorte, die mit dem Präfix `http://` angegeben werden. Sie können auf eine Ressource nicht mit einer HTTP-URL verweisen. Beispiel: Das Übergeben von `http://mysite/myjar.jar` als JAR-Parameter bewirkt, dass der Cluster fehlschlägt.

Verweisen Sie mit einem ungültigen Namensformat auf einen Amazon-S3-Bucket?

Wenn Sie versuchen, einen Bucket-Namen wie „amzn-s3-demo-bucket1.1“ mit Amazon EMR zu verwenden, schlägt Ihr Cluster fehl, weil Amazon EMR verlangt, dass Bucket-Namen gültige RFC 2396-Hostnamen sind; der Name darf nicht mit einer Zahl enden. Um die Anforderungen von Hadoop zu erfüllen, dürfen Namen von mit Amazon EMR verwendeten Amazon-S3-Buckets darüber hinaus nur Kleinbuchstaben, Zahlen, Punkte (.) und Bindestriche (-) enthalten. Weitere Informationen zum Formatieren von Amazon-S3-Bucket-Namen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Benutzerhandbuch für Amazon Simple Storage Service.

Haben Sie Probleme beim Laden von Daten in oder aus Amazon S3?

Amazon S3 ist die beliebteste Ein- und Ausgabequelle für Amazon EMR. Ein häufiger Fehler besteht darin, Amazon S3 so zu behandeln wie ein typisches Dateisystem. Es gibt Unterschiede zwischen Amazon S3 und einem Dateisystem, die Sie berücksichtigen müssen, wenn Sie Ihren Cluster ausführen.

- Wenn ein interner Fehler in Amazon S3 auftritt, muss Ihre Anwendung diesen problemlos behandeln und die Operation wiederholen.
- Wenn Aufrufe in Amazon S3 zu lange dauern, muss Ihre Anwendung die Häufigkeit der Amazon-S3-Aufrufe ggf. reduzieren.
- Das Auflisten aller Objekte in einem Amazon-S3-Bucket ist ein teurer Aufruf. Ihre Anwendung sollte die Anzahl solcher Aufrufe minimieren.

Es gibt mehrere Möglichkeiten, wie Sie die Interaktion Ihres Cluster mit Amazon S3 verbessern können.

- Starten Sie Ihren Cluster mit der neuesten Version von Amazon EMR.
- Verwenden Sie S3DistCp , um Objekte in und aus Amazon S3 zu verschieben. S3 DistCp implementiert Fehlerbehandlung, Wiederholungsversuche und Back-offs, um die Anforderungen von Amazon S3 zu erfüllen. Weitere Informationen finden Sie unter [Verteilte Kopie](#) mit S3. DistCp
- Entwickeln Sie Ihre Anwendung mit letztendlicher Datenkonsistenz im Blick. Verwenden Sie HDFS für das Zwischenspeichern von Daten, während der Cluster ausgeführt wird, und Amazon S3 nur für die Eingabe der Ausgangsdaten und für die Ausgabe der Endergebnisse.
- Wenn Ihre Cluster einen Commit für mindestens 200 Transaktionen pro Sekunde in Amazon S3 [contact support](#) durchführen, wenden Sie sich an den Support, um Ihren Bucket auf größere Transaktionen pro Sekunde vorzubereiten. Ziehen Sie dazu die unter [Tipps und Tricks zur Leistung von Amazon S3](#) beschriebenen Strategien in Erwägung.
- Legen Sie die Hadoop-Konfigurationseinstellung "io.file.buffer.size" auf "65536" fest. Diese bewirkt, dass Hadoop weniger Zeit damit verbringt, Amazon-S3-Objekte zu durchsuchen.
- Überlegen Sie sich, ob Sie das speculative-Execution-Feature in Hadoop deaktivieren, wenn Ihr Cluster Probleme mit der gleichzeitigen Ausführung von Amazon S3 hat. Diese Vorgehensweise ist auch bei der Problembehandlung eines langsamen Clusters nützlich. Sie tun dies, indem Sie die `mapreduce.map.speculative`- und die `mapreduce.reduce.speculative`-Eigenschaften auf `false` festlegen. Wenn Sie einen Cluster starten, können Sie diese Werte mithilfe der `mapred-env`-Konfigurationsklassifizierung festlegen. Weitere Informationen finden Sie unter [Konfigurieren von Anwendungen](#) in den Amazon EMR-Versionshinweisen.
- Wenn Sie einen Hive-Cluster ausführen, finden Sie weitere Informationen unter [Haben Sie Probleme beim Laden von Daten in oder aus Amazon S3 in Hive?](#).

Weitere Informationen finden Sie unter [Bewährte Methoden für Amazon-S3-Fehler](#) im Benutzerhandbuch von Amazon Simple Storage Service.

Berechtigungsfehler bei Amazon EMR-Clustervorgängen

Die folgenden Fehler treten häufig im Zusammenhang mit Berechtigungen oder Anmeldeinformationen auf.

Themen

- [Haben Sie für SSH die korrekten Anmeldeinformationen angegeben?](#)
- [Wenn Sie IAM verwenden, haben Sie die richtigen EC2 Amazon-Richtlinien festgelegt?](#)

Haben Sie für SSH die korrekten Anmeldeinformationen angegeben?

Wenn Sie keine SSH-Verbindung zum Master-Knoten herstellen können, gibt es höchstwahrscheinlich ein Problem mit Ihren Anmeldeinformationen.

Prüfen Sie zunächst, ob die PEM-Datei mit dem SSH-Schlüssel über die entsprechenden Berechtigungen verfügt. Verwenden Sie `chmod`, um die Berechtigungen für Ihre .PEM-Datei zu ändern. Im folgenden Beispiel würden Sie `mykey.pem` durch den Namen Ihrer eigenen PEM-Datei ersetzen.

```
chmod og-rwx mykey.pem
```

Die zweite Fehlerquelle besteht darin, dass Sie nicht das Schlüsselpaar verwenden, das Sie beim Erstellen des Clusters angegeben haben. Dies passiert schnell, falls Sie mehrere Schlüsselpaare erstellt haben. Prüfen Sie die Cluster-Details in der Amazon-EMR-Konsole (oder verwenden Sie die Option `--describe` in der CLI). Stellen Sie fest, ob der Name des Schlüsselpaares mit dem bei der Erstellung des Clusters angegebenen übereinstimmt.

Nachdem Sie überprüft haben, ob Sie das richtige Schlüsselpaar und die korrekten Berechtigungen für die PEM-Datei verwendet haben, können Sie den folgenden Befehl nutzen, um eine SSH-Verbindung mit dem Hauptknoten herzustellen. Ersetzen Sie `mykey.pem` durch den Namen Ihrer PEM-Datei und `hadoop@ec2-01-001-001-1.compute-1.amazonaws.com` durch den öffentlichen DNS-Namen des Hauptknotens (über die Option `--describe` in der CLI und über die Amazon-EMR-Konsole abrufbar).

Important

Sie müssen den Anmeldenamen `hadoop` verwenden, wenn Sie eine Verbindung mit dem Amazon-EMR-Cluster-Knoten herstellen. Andernfalls kann eine Fehlermeldung wie `Server refused our key` angezeigt werden.

```
ssh -i mykey.pem hadoop@ec2-01-001-001-1.compute-1.amazonaws.com
```

Weitere Informationen finden Sie unter [Stellen Sie mithilfe von SSH eine Connect zum primären Knoten des Amazon EMR-Clusters her](#).

Wenn Sie IAM verwenden, haben Sie die richtigen EC2 Amazon-Richtlinien festgelegt?

Da Amazon EMR EC2 Instances als Knoten verwendet, müssen für Benutzer von Amazon EMR auch bestimmte EC2 Amazon-Richtlinien eingerichtet sein, damit Amazon EMR diese Instances im Namen eines Benutzers verwalten kann. Wenn Sie nicht über die erforderlichen Berechtigungen verfügen, gibt Amazon EMR die folgende Fehlermeldung zurück: „Das Konto ist nicht zum Anrufen autorisiert EC2.“

Weitere Informationen zu den EC2 Amazon-Richtlinien, die Ihr IAM-Konto für die Ausführung von Amazon EMR einrichten muss, finden Sie unter [Funktionsweise von Amazon EMR mit IAM](#)

Hive-Cluster-Fehler

Den Grund für einen Hive-Fehler finden Sie in der Regel in der Datei `syslog`, auf die Sie im Bereich Steps (Schritte) zugreifen können. Wenn Sie das Problem nicht ermitteln können, sehen Sie sich die Fehlermeldung für die versuchte Hadoop-Aufgabe an. Erstellen Sie einen Link dahin im Abschnitt Task Attempts (Aufgaben-Versuche).

Die folgenden Fehler treten häufig bei Hive-Clustern auf.

Themen

- [Verwenden Sie die neueste Version von Hive?](#)
- [Ist im Hive-Skript ein Syntaxfehler aufgetreten?](#)
- [Ist ein interaktiv ausgeführter Auftrag fehlgeschlagen?](#)

- [Haben Sie Probleme beim Laden von Daten in oder aus Amazon S3 in Hive?](#)

Verwenden Sie die neueste Version von Hive?

Die neueste Version von Hive verfügt über alle aktuellen Patches und Fehlerbehebungen und kann Ihr Problem lösen.

Ist im Hive-Skript ein Syntaxfehler aufgetreten?

Wenn ein Schritt fehlschlägt, sehen Sie sich die Datei `stdout` der Protokolle für den Schritt an, die das Hive-Skript ausgeführt hat. Wenn der Fehler nicht vorhanden ist, sehen Sie sich die Datei `syslog` der Aufgabenprotokolle für die versuchte Aufgabe an, die fehlgeschlagen ist. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Ist ein interaktiv ausgeführter Auftrag fehlgeschlagen?

Wenn Sie Hive interaktiv auf dem Master-Knoten ausführen und der Cluster fehlschlägt, sehen Sie sich die Einträge `syslog` im Aufgabenprotokoll für die fehlgeschlagene Aufgabe an. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Haben Sie Probleme beim Laden von Daten in oder aus Amazon S3 in Hive?

Falls Sie Probleme mit dem Zugriff auf Daten in Amazon S3 haben, überprüfen Sie zuerst die möglichen Ursachen, die in [Haben Sie Probleme beim Laden von Daten in oder aus Amazon S3?](#) aufgeführt sind. Wenn keines dieser Probleme die Ursache ist, ziehen Sie die folgenden, für Hive spezifischen Optionen in Betracht.

- Stellen Sie sicher, dass Sie die neueste Version von Hive verwenden, die über alle aktuellen Patches und Fehlerbehebungen verfügt, die Ihr Problem lösen können. Weitere Informationen finden Sie unter [Apache Hive](#).
- Wenn Sie `INSERT OVERWRITE` verwenden, müssen Sie die Inhalte des Amazon-S3-Buckets oder -Ordners auflisten. Dies ist eine teure Operation. Wenn möglich, optimieren Sie den Pfad manuell die vorhandenen Objekte von Hive auflisten und löschen zu lassen.
- Wenn Sie ältere Versionen als Amazon EMR 5.0 verwenden, können Sie den folgenden Befehl in HiveQL ausführen, um die Ergebnisse einer Amazon-S3-Auflistungsoperation vorab lokal auf dem Cluster zwischenzuspeichern:

```
set hive.optimize.s3.query=true;
```

- Verwenden Sie statische Partitionen, wenn möglich.
- In einigen Versionen von Hive und Amazon EMR ist es möglich, dass mit ALTER TABLES ein Fehler auftritt, da die Tabelle an einem anderen Ort gespeichert ist, als von Hive erwartet wird. Die Lösung ist, Folgendes in `/home/hadoop/conf/core-site.xml` hinzuzufügen oder zu aktualisieren:

```
<property>
  <name>fs.s3n.endpoint</name>
  <value>s3.amazonaws.com</value>
</property>
```

VPC-Fehler bei Amazon EMR-Clusteroperationen

Die folgenden Fehler treten häufig bei der VPC-Konfiguration in Amazon EMR auf.

Themen

- [Ungültige Subnetzkonfiguration](#)
- [Fehlende DHCP-Optionsliste](#)
- [Berechtigungsfehler](#)
- [Fehler, die zu START_FAILED führen](#)
- [Cluster und kann nicht gestartet werden Terminated with errors NameNode](#)

Ungültige Subnetzkonfiguration

Auf der Seite Cluster Details (Cluster-Details) im Feld Status sehen Sie eine Fehlermeldung wie folgende:

The subnet configuration was invalid: Cannot find route to InternetGateway in main RouteTable *rtb-id* for vpc *vpc-id*.

Um dieses Problem zu lösen, müssen Sie ein Internet-Gateway erstellen und Ihre VPC anfügen. Weitere Informationen finden Sie unter [Hinzufügen eines Internet-Gateways zu Ihrer VPC](#).

Alternativ stellen Sie sicher, dass Sie Ihre VPC mit Enable DNS resolution (DNS-Auflösung aktivieren) und Enable DNS hostname support (DNS-Hostnamen-Unterstützung aktivieren) aktiviert konfiguriert haben. Weitere Informationen finden Sie unter [Verwenden von DNS in Ihrer VPC](#).

Fehlende DHCP-Optionsliste

Sie sehen einen Schrittfehler im Cluster-Systemprotokoll (syslog) mit einer Fehlermeldung ähnlich der folgenden:

```
ERROR org.apache.hadoop.security.UserGroupInformation
(main): PrivilegedActionException as:hadoop (auth:SIMPLE)
cause:java.io.IOException:
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

or

```
ERROR org.apache.hadoop.streaming.StreamJob (main): Error Launching job :
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

Um dieses Problem zu lösen, müssen Sie eine VPC mit einer DHCP-Optionsliste konfigurieren, deren Parameter auf die folgenden Werte festgelegt sind:

Note

Wenn Sie die Region AWS GovCloud (US-West) verwenden, setzen Sie domain-name auf **us-gov-west-1.compute.internal** anstelle des im folgenden Beispiel verwendeten Werts.

- domain-name = **ec2.internal**

Verwenden Sie **ec2.internal**, wenn Ihre Region USA Ost (Nord-Virginia) ist. Verwenden Sie für andere Regionen. *region-name* **.compute.internal** Verwenden Sie zum Beispiel in us-west-2 domain-name=**us-west-2.compute.internal**.

- domain-name-servers = **AmazonProvidedDNS**

Weitere Informationen finden Sie unter [DHCP-Options-Sets](#).

Berechtigungsfehler

Ein Fehler im stderr-Protokoll für einen Schritt gibt an, dass eine Amazon-S3-Ressource nicht über die entsprechenden Berechtigungen verfügt. Dies ist ein Fehler 403, der wie folgt aussieht:


```
Exception in thread "main" com.amazonaws.services.s3.model.AmazonS3Exception: Access
Denied (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request
ID: REQUEST_ID)
```

Wenn der auf gesetzte ActionOnFailure ist `TERMINATE_JOB_FLOW`, würde dies dazu führen, dass der Cluster mit dem Status, `SHUTDOWN_COMPLETED_WITH_ERRORS` beendet wird.

Möglichkeiten, um dieses Problem zu beheben, sind beispielsweise:

- Wenn Sie eine Amazon-S3-Bucket-Richtlinie in einer VPC verwenden, stellen Sie sicher, dass der Zugriff auf alle Buckets ermöglicht wird. Erstellen Sie dazu einen VPC-Endpoint und wählen Sie beim Erstellen des Endpunkts Alle zulassen unter der Option „Richtlinie“ aus.
- Stellen Sie sicher, dass alle Richtlinien im Zusammenhang mit S3-Ressourcen die VPC umfassen, in der Sie den Cluster starten.
- Führen Sie den folgenden Befehl über Ihren Cluster aus, um zu überprüfen, ob Sie auf den Bucket zugreifen können.

```
hadoop fs -copyToLocal s3://path-to-bucket /tmp/
```

- Sie können spezifischere Debugging-Informationen abrufen, indem Sie den Parameter `log4j.logger.org.apache.http.wire` in der Datei `DEBUG-Datei` im Cluster auf `/home/hadoop/conf/log4j.properties` festlegen. Sie können die `stderr`-Protokolldatei prüfen, nachdem Sie versucht haben, über den Cluster auf den Bucket zuzugreifen. Die Protokolldatei enthält detaillierte Informationen:

```
Access denied for getting the prefix for bucket - us-west-2.elasticmapreduce with
path samples/wordcount/input/
15/03/25 23:46:20 DEBUG http.wire: >> "GET /?prefix=samples%2Fwordcount%2Finput
%2F&delimiter=%2F&max-keys=1 HTTP/1.1[\r][\n]"
15/03/25 23:46:20 DEBUG http.wire: >> "Host: us-
west-2.elasticmapreduce.s3.amazonaws.com[\r][\n]"
```

Fehler, die zu **START_FAILED** führen

Vor AMI 3.7.0, für die VPCs ein Hostname angegeben wurde, ordnet Amazon EMR die internen Hostnamen des Subnetzes benutzerdefinierten Domainadressen wie folgt zu:

`ip-X.X.X.X.customdomain.com.tld` Wenn beispielsweise der Hostname `ip-10.0.0.10` lautet und die Domainnamenoption der VPC auf `customdomain.com` festgelegt ist, wird von Amazon

EMR der Hostname `ip-10.0.1.0.customdomain.com` zugeordnet. Ein Eintrag wird in `/etc/hosts` hinzugefügt, um den Hostnamen in `10.0.0.10` aufzulösen. Dieses Verhalten wird ab AMI 3.7.0 geändert. Jetzt erkennt Amazon EMR die DHCP-Konfiguration der VPC vollständig an. Bislang konnten Kunden eine Zuweisung des Hostnamens auch mit einer Bootstrap-Aktion angeben.

Wenn Sie dieses Verhalten beibehalten möchten, müssen Sie die Einrichtung der DNS- und Weiterleitungsauflösung angeben, die Sie für die benutzerdefinierte Domain benötigen.

Cluster und kann nicht gestartet werden **Terminated with errors** NameNode

Beim Starten eines EMR-Clusters in einer VPC, die einen benutzerdefinierten DNS-Domainnamen verwendet, tritt bei Ihrem Cluster möglicherweise ein Fehler mit der folgende Fehlermeldung in der Konsole auf:

```
Terminated with errors On the master instance(instance-id), bootstrap action 1
returned a non-zero return code
```

Der Fehler ist darauf zurückzuführen, dass der Start NameNode nicht möglich ist. Dies führt dazu, dass der folgende Fehler in den NameNode Protokollen gefunden wird, deren Amazon S3 S3-URI die folgende Form hats3: `//amzn-s3-demo-bucket/logs/cluster-id/daemons/master instance-id/hadoop-hadoop-namenode-master node hostname.log.gz`:

```
2015-07-23 20:17:06,266 WARN
    org.apache.hadoop.hdfs.server.namenode.FSNamesystem (main): Encountered
exception
    loading fsimage java.io.IOException: NameNode is not formatted.
    at

org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:212)
    at

org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:1020)
    at

org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:739)
    at
    org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:537)
    at
    org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:596)

at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:765)
at
```

```
org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:749)
at
org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1441)
at
org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1507)
```

Dies ist auf ein potenzielles Problem zurückzuführen, bei dem eine EC2 Instance mehrere Sätze vollqualifizierter Domännennamen haben kann, wenn EMR-Cluster in einer VPC gestartet werden, die sowohl einen vom Benutzer AWS bereitgestellten DNS-Server als auch einen benutzerdefinierten DNS-Server verwendet. Wenn der vom Benutzer bereitgestellte DNS-Server keine Zeigerdatensätze (PTR) für die A-Datensätze bereitstellt, die zum Angeben von Knoten in einem EMR-Cluster dienen, können die so konfigurierten Cluster nicht starten. Die Lösung besteht darin, für jeden A-Datensatz, der erstellt wird, wenn eine EC2 Instance in einem der Subnetze der VPC gestartet wird, einen PTR-Datensatz hinzuzufügen.

Fehler beim Streaming von Amazon EMR-Clustern

Sie können in der Regel die Ursache für einen Streaming-Fehler in einer syslog-Datei finden. Erstellen Sie einen Link dahin im Abschnitt Steps (Schritte).

Die folgenden Fehler treten häufig bei Streaming-Clustern auf.

Themen

- [Werden Daten an den Mapper im falschen Format gesendet?](#)
- [Gibt es eine Zeitüberschreitung bei der Skriptausführung?](#)
- [Werden ungültige Streaming-Argumente übergeben?](#)
- [Wurde Ihr Skript mit einem Fehler beendet?](#)

Werden Daten an den Mapper im falschen Format gesendet?

Suchen Sie in der syslog-Datei nach einer Fehlermeldung über einen fehlgeschlagenen Aufgabenversuch in den Protokolldateien der Aufgabenversuche, um dies zu überprüfen. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Gibt es eine Zeitüberschreitung bei der Skriptausführung?

Die standardmäßige Zeitbeschränkung für ein Mapper- oder Reducer-Skript beträgt 600 Sekunden. Wenn Ihr Skript mehr Zeit benötigt, schlägt der Aufgabenversuch fehl. Suchen Sie in der

syslog-Datei nach einem fehlgeschlagenen Aufgabenversuch in den Protokolldateien der Aufgabenversuche, um dies zu überprüfen. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Sie können die Zeitbeschränkung ändern, indem Sie einen neuen Wert für die Konfigurationseinstellung `mapred.task.timeout` festlegen. Diese Einstellung gibt die Anzahl von Millisekunden an, nach der Amazon EMR eine Aufgabe beendet, die keine Eingabe gelesen, Ausgabe geschrieben oder ihre Status-Zeichenfolge nicht aktualisiert hat. Sie können diesen Wert aktualisieren, indem Sie ein zusätzliches Streaming-Argument `-jobconf mapred.task.timeout=800000` übergeben.

Werden ungültige Streaming-Argumente übergeben?

Hadoop-Streaming unterstützt nur die folgenden Argumente. Wenn Sie andere als die unten aufgeführten Argumente übergeben, schlägt der Cluster fehl.

```
-blockAutoGenerateCacheFiles
-cacheArchive
-cacheFile
-cmdenv
-combiner
-debug
-input
-inputformat
-inputreader
-jobconf
-mapper
-numReduceTasks
-output
-outputformat
-partitioner
-reducer
-verbose
```

Darüber hinaus erkennt Hadoop-Streaming nur in Java-Syntax übergebene Argumente, also mit einem vorangestellten einzelnen Bindestrich. Wenn Argumente mit vorangestelltem doppeltem Bindestrich übergeben werden, schlägt der Cluster fehl.

Wurde Ihr Skript mit einem Fehler beendet?

Wenn Ihr Mapper- oder Reducer-Skript mit einem Fehler beendet wird, können Sie den Fehler in der `stderr`-Datei des fehlgeschlagenen Aufgabenversuchs in den Protokolldateien der Aufgabenversuche ermitteln. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Amazon EMR: Fehler beim benutzerdefinierten JAR-Cluster

Die folgenden Fehler treten häufig bei benutzerdefinierten JAR-Clustern auf.

Themen

- [Löst Ihr JAR-Cluster vor dem Erstellen eines Auftrags eine Ausnahme aus?](#)
- [Tritt auf Ihrem JAR-Cluster ein Fehler in einer Map-Aufgabe auf?](#)

Löst Ihr JAR-Cluster vor dem Erstellen eines Auftrags eine Ausnahme aus?

Wenn das Hauptprogramm des benutzerdefinierten JAR-Clusters einen Ausnahmefehler beim Erstellen des Hadoop-Auftrags ausgibt, sehen Sie sich am besten die Datei `syslog` der Schrittprotokolle an. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Tritt auf Ihrem JAR-Cluster ein Fehler in einer Map-Aufgabe auf?

Wenn Ihr benutzerdefinierte JAR-Cluster und Mapper einen Ausnahmefehler beim Verarbeiten von Eingabedaten ausgibt, sehen Sie sich am besten die Datei `syslog` der Aufgabenversuchsprotokolle an. Weitere Informationen finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Fehler bei Amazon EMR AWS GovCloud (US-West)

Die Region AWS GovCloud (US-West) unterscheidet sich von anderen Regionen in Bezug auf Sicherheit, Konfiguration und Standardeinstellungen. Verwenden Sie daher die folgende Checkliste, um Amazon EMR-Fehler zu beheben, die für die Region AWS GovCloud (USA West) spezifisch sind, bevor Sie allgemeinere Empfehlungen zur Fehlerbehebung verwenden.

- Überprüfen Sie, ob Ihre IAM-Rollen ordnungsgemäß konfiguriert sind. Weitere Informationen finden Sie unter [Konfigurieren Sie IAM-Servicerollen für Amazon-EMR-Berechtigungen für AWS Services und Ressourcen](#).

- Stellen Sie sicher, dass Ihre VPC-Konfiguration die Parameter `resolution/hostname` DNS-Unterstützung, Internet Gateway und DHCP-Optionssatz korrekt konfiguriert hat. Weitere Informationen finden Sie unter [VPC-Fehler bei Amazon EMR-Clusteroperationen](#).

Wenn diese Schritte das Problem nicht lösen, fahren Sie mit den Schritten zur Problembeseitigung allgemeiner Amazon-EMR-Fehler fort. Weitere Informationen finden Sie unter [Sammlungen häufiger Fehler in Amazon EMR](#).

Finden Sie einen fehlenden Cluster

Wenn Ihr Cluster in der Konsolenliste oder `ListClusters`-API fehlt, überprüfen Sie Folgendes:

- Vergewissern Sie sich, dass das Alter des Clusters ab dem Zeitpunkt der Fertigstellung weniger als zwei Monate beträgt. Amazon EMR bewahrt Metadateninformationen für abgeschlossene Cluster zwei Monate lang kostenlos auf. Sie können abgeschlossene Cluster nicht aus der Konsole löschen. Stattdessen löscht Amazon EMR abgeschlossene Cluster automatisch nach zwei Monaten.
- Bestätigen Sie, dass Sie über Rollenberechtigungen zum Anzeigen des Clusters verfügen.
- Vergewissern Sie sich, dass Sie dort, AWS-Region wo sich der Cluster befindet, dieselbe Ansicht sehen.

Fehlerbehebung bei einem Amazon EMR-Cluster, der mit einem Fehlercode ausgefallen ist

In diesem Abschnitt werden Sie durch den Vorgang zur Fehlerbehebung eines Cluster geführt, der ausgefallen ist. Das bedeutet, dass der Cluster mit einem Fehlercode beendet wurde.

Note

Wenn ein EMR-Cluster mit einem Fehler beendet wird, werden ein Fehlercode und eine Fehlermeldung `ListClusters` APIs zurückgegeben. `DescribeCluster` Bei einigen Clusterfehlern kann Ihnen das `ErrorDetail`-Datenarray bei der Behebung des Fehlers helfen. Weitere Informationen finden Sie unter [Fehlercodes mit ErrorDetail Informationen in Amazon EMR](#).

Wenn Ihr Cluster ausgeführt wird, aber lange braucht, bis Ergebnisse zurückgegeben werden, finden Sie unter [Fehlerbehebung bei einem langsamen Amazon EMR-Cluster](#).

Themen

- [Schritt 1: Sammeln Sie Daten über das Problem mit dem Amazon EMR-Cluster](#)
- [Schritt 2: Die Umgebung prüfen](#)
- [Schritt 3: Die letzte Statusänderung überprüfen](#)
- [Schritt 4: Untersuchen Sie die Amazon EMR-Protokolldateien](#)
- [Schritt 5: Testen Sie den Amazon EMR-Cluster Schritt für Schritt](#)

Schritt 1: Sammeln Sie Daten über das Problem mit dem Amazon EMR-Cluster

Der erste Schritt bei der Fehlerbehebung bei einem Cluster besteht darin, Informationen darüber zu sammeln, was schief gelaufen ist, sowie über den aktuellen Status und die Konfiguration des Clusters. Diese Informationen werden in den folgenden Schritten verwendet, um mögliche Ursachen des Problems zu bestätigen oder auszuschließen.

Definieren des Problems

Eine klare Definition des Problems ist der erste Ausgangspunkt. Einige Fragen, die Sie sich stellen sollten:

- Was habe ich erwartet? Was ist stattdessen passiert?
- Wann ist dieses Problem zum ersten Mal aufgetreten? Wie oft ist es seitdem passiert?
- Hat sich etwas an der Konfiguration oder Ausführung meines Clusters geändert?

Cluster-Details

Die folgenden Clusterdetails sind hilfreich, um Probleme aufzuspüren. Weitere Informationen zum Sammeln dieser Informationen finden Sie unter [Status und Details des Amazon EMR-Clusters anzeigen](#).

- Die Cluster-ID. (Wird auch als Job-Flow-Identifizier bezeichnet.)
- AWS-Region und Availability Zone, in der der Cluster gestartet wurde.

- Status des Clusters, einschließlich Details zur letzten Statusänderung.
- Typ und Anzahl der für die Master-, Core- und Task-Knoten angegebenen EC2 Instanzen.

Schritt 2: Die Umgebung prüfen

Amazon EMR wird als Teil eines Ökosystems von Web-Services und Open-Source-Software betrieben. Faktoren, die sich auf diese Abhängigkeiten auswirken, können die Leistung von Amazon EMR beeinträchtigen.

Themen

- [Prüfen auf Service-Ausfälle](#)
- [Prüfen auf Nutzungsgrenzen](#)
- [Überprüfen der Version](#)
- [Prüfen der Amazon-VPC-Subnetzkonfiguration](#)

Prüfen auf Service-Ausfälle

Amazon EMR verwendet intern mehrere Amazon Web Services. Es betreibt virtuelle Server auf Amazon EC2, speichert Daten und Skripts auf Amazon S3 und meldet Metriken an CloudWatch. Ereignisse, die diese Services stören, sind selten – wenn sie jedoch auftreten, können sie zu Problemen in Amazon EMR führen.

Überprüfen Sie die [Übersicht zum Servicestatus](#), bevor Sie fortfahren. Prüfen Sie in der Region, in der Sie Ihren Cluster gestartet haben, ob es bei einem dieser Services zu Störungen gekommen ist.

Prüfen auf Nutzungsgrenzen

Wenn Sie einen großen Cluster starten, viele Cluster gleichzeitig gestartet haben oder wenn Sie ein Benutzer sind, der einen Cluster AWS-Konto mit anderen Benutzern teilt, ist der Cluster möglicherweise ausgefallen, weil Sie ein AWS Service-Limit überschritten haben.

Amazon EC2 begrenzt die Anzahl der virtuellen Server-Instances, die in einer einzelnen AWS Region ausgeführt werden, auf 20 On-Demand-Instances oder Reserved Instances. Wenn Sie einen Cluster mit mehr als 20 Knoten starten oder einen Cluster starten, der dazu führt, dass die Gesamtzahl der AWS-Konto auf Ihrem Computer aktiven EC2 Instances 20 überschreitet, kann der Cluster nicht alle benötigten EC2 Instances starten und schlägt möglicherweise fehl. In diesem Fall gibt Amazon EMR einen EC2 QUOTA EXCEEDED-Fehler zurück. Sie können beantragen, dass die Anzahl der EC2

Instances, die Sie auf Ihrem Konto ausführen können, AWS erhöht wird, indem Sie einen [Antrag auf Erhöhung des EC2 Amazon-Instance-Limits](#) einreichen.

Eine weitere Sache, die dazu führen kann, dass Sie Ihre Nutzungslimits überschreiten, ist die Verzögerung zwischen der Beendigung eines Clusters und der Freigabe aller seiner Ressourcen. Je nach Konfiguration kann es bis zu 5–20 Minuten dauern, bis ein Cluster vollständig beendet ist und zugewiesene Ressourcen freigibt. Wenn Sie beim Versuch, einen Cluster zu starten, die Fehlermeldung EC2 QUOTA EXCEEDED erhalten, kann es daran liegen, dass Ressourcen eines kürzlich beendeten Clusters noch nicht zur Verfügung stehen. In diesem Fall können Sie entweder [beantragen, dass Ihr EC2 Amazon-Kontingent erhöht wird](#), oder Sie können zwanzig Minuten warten und den Cluster neu starten.

Amazon S3 begrenzt die Anzahl der auf einem Konto erstellten Buckets auf 100. Wenn Ihr Cluster einen neuen Bucket erstellt, der dieses Limit überschreitet, schlägt die Bucket-Erstellung fehl und kann dazu führen, dass der Cluster fehlschlägt.

Überprüfen der Version

Vergleichen Sie die Versionsbezeichnung, die Sie zum Start des Clusters verwendet haben, mit der aktuellen Amazon-EMR-Version. Jede Version von Amazon EMR beinhaltet Verbesserungen, wie z. B. neue Anwendungen, Features, Patches und Fehlerbehebungen. Das Problem, das Ihren Cluster betrifft, wurde in der aktuellen Version möglicherweise bereits behoben. Führen Sie Ihren Cluster wenn möglich mit der aktuellen Version erneut aus.

Prüfen der Amazon-VPC-Subnetzkonfiguration

Wenn Ihr Cluster in einem Amazon VPC-Subnetz gestartet wurde, muss das Subnetz wie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#) beschrieben konfiguriert werden. Überprüfen Sie außerdem, ob das Subnetz, in dem Sie den Cluster starten, über genügend freie elastische IP-Adressen verfügt, um jedem Knoten im Cluster eine zuzuweisen.

Schritt 3: Die letzte Statusänderung überprüfen

Die letzte Statusänderung gibt Aufschluss darüber, welches Ereignis bei der letzten Statusänderung des Clusters aufgetreten ist. Häufig lassen sich Informationen gewinnen, die Hinweise darauf geben, welcher Fehler auftrat, als sich der Cluster-Status in FAILED änderte. Wenn Sie beispielsweise einen Streaming-Cluster starten und einen Ausgabespeicherort angeben, der in Amazon S3 bereits vorhanden ist, tritt beim Cluster ein Fehler auf und die letzte Statusänderung weist darauf hin, dass das Streaming-Ausgabeverzeichnis bereits vorhanden ist.

Sie können den Wert der letzten Statusänderung über die Konsole ermitteln, indem Sie den Detailbereich für den Cluster über die Befehlszeilenschnittstelle (CLI) mit dem Argument `list-steps` oder `describe-cluster` bzw. über die API mit den Aktionen `DescribeCluster` und `ListSteps` anzeigen. Weitere Informationen finden Sie unter [Status und Details des Amazon EMR-Clusters anzeigen](#).

Schritt 4: Untersuchen Sie die Amazon EMR-Protokolldateien

Der nächste Schritt besteht darin, die Protokolldateien zu untersuchen, um einen Fehlercode oder einen anderen Hinweis auf das Problem zu finden, das in Ihrem Cluster aufgetreten ist. Informationen zu den verfügbaren Protokolldateien, wo sie zu finden sind und wie Sie sie anzeigen können, finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Es kann einige Nachforschungen erfordern, um herauszufinden, was passiert ist. Hadoop führt die Arbeit der Aufträge in Aufgabenversuchen auf verschiedenen Knoten im Cluster aus. Amazon EMR kann spekulative Aufgabenversuche initiieren und die anderen Aufgabenversuche beenden, die nicht zuerst abgeschlossen werden. Dadurch werden umfangreiche Aktivitäten generiert, die in den Controller-, Stderr- und Syslog-Protokolldateien protokolliert werden. Darüber hinaus werden mehrere Aufgaben gleichzeitig ausgeführt, aber eine Protokolldatei kann die Ergebnisse nur linear anzeigen.

Überprüfen Sie zunächst die Bootstrap-Aktionsprotokolle auf Fehler oder unerwartete Konfigurationsänderungen beim Start des Clusters. Suchen Sie anschließend in den Schrittprotokollen nach Hadoop-Aufträgen, die als Teil eines fehlerhaften Schritts gestartet wurden. Untersuchen Sie die Hadoop-Auftragsprotokolle, um die fehlgeschlagenen Aufgabenversuche zu identifizieren. Das Protokoll der Aufgabenversuche wird Details darüber enthalten, was zum Fehlschlagen eines Aufgabenversuchs geführt hat.

In den folgenden Abschnitten wird erläutert, wie die verschiedenen Protokolldateien verwendet werden, um Fehler in Ihrem Cluster zu identifizieren.

Die Bootstrap-Aktionsprotokolle überprüfen

Bootstrap-Aktionen führen Skripts auf dem Cluster aus, während dieser gestartet wird. Sie werden häufig verwendet, um zusätzliche Software auf dem Cluster zu installieren oder um Konfigurationseinstellungen gegenüber den Standardwerten zu ändern. Die Überprüfung dieser Protokolle kann Aufschluss über Fehler geben, die bei der Einrichtung des Clusters aufgetreten sind, sowie über Änderungen der Konfigurationseinstellungen, die sich auf die Leistung auswirken könnten.

Die Schrittprotokolle überprüfen

Es gibt vier Arten von Schrittprotokollen.

- **Controller** – Enthält von Amazon EMR (Amazon EMR) generierte Dateien, die auf Fehler zurückzuführen sind, die bei der Ausführung Ihres Schritts aufgetreten sind. Wenn Ihr Schritt beim Laden fehlschlägt, finden Sie den Stack-Trace in diesem Protokoll. Fehler beim Laden oder Zugreifen auf Ihre Anwendung werden hier häufig beschrieben, ebenso wie Fehler in der fehlenden Mapper-Datei.
- **stderr** – Enthält Fehlermeldungen, die bei der Verarbeitung des Schritts aufgetreten sind. Fehler beim Laden von Anwendungen werden hier häufig beschrieben. Dieses Protokoll enthält manchmal einen Stack-Trace.
- **stdout** – Enthält den Status, der von Ihren ausführbaren Mapper- und Reducer-Dateien generiert wurde. Fehler beim Laden von Anwendungen werden hier häufig beschrieben. Dieses Protokoll enthält manchmal Anwendungsfehlermeldungen.
- **syslog** – Enthält Protokolle von Software, die nicht von Amazon stammt, wie Apache und Hadoop. Streaming-Fehler werden hier häufig beschrieben.

Überprüfen Sie stderr auf offensichtliche Fehler. Wenn stderr eine kurze Liste von Fehlern anzeigt, wurde der Schritt schnell beendet und es wurde ein Fehler ausgelöst. Dies wird meistens durch einen Fehler in den Mapper- und Reducer-Anwendungen verursacht, die im Cluster ausgeführt werden.

Untersuchen Sie die letzten Zeilen von Controller und Syslog auf Hinweise auf Fehler oder Ausfälle. Folgen Sie allen Hinweisen zu fehlgeschlagenen Aufgaben, insbesondere wenn dort „Auftrag fehlgeschlagen“ steht.

Die Aufgabenversuchsprotokolle überprüfen

Wenn die vorherige Analyse der Schrittprotokolle eine oder mehrere fehlgeschlagene Aufgaben ergeben hat, suchen Sie in den Protokollen der entsprechenden Aufgabenversuche nach detaillierteren Fehlerinformationen.

Schritt 5: Testen Sie den Amazon EMR-Cluster Schritt für Schritt

Eine nützliche Strategie zum Nachverfolgen der Ursache für einen Fehler besteht darin, den Cluster neu zu starten und die Schritte einzeln auszuführen. So können Sie die Ergebnisse für jeden Schritt überprüfen, bevor Sie die Verarbeitung des nächsten Schritts starten, und erhalten die Möglichkeit,

einen fehlgeschlagenen Schritt zu korrigieren und erneut auszuführen. Dies hat den Vorteil, dass Sie Ihre Eingabedaten nur einmal laden müssen.

So testen Sie den Cluster Schritt für Schritt

1. Starten Sie einen neuen Cluster mit aktiviertem Keepalive und Beendigungsschutz. Keepalive sorgt dafür, dass der Cluster weiterhin ausgeführt wird, nachdem er alle ausstehenden Schritte verarbeitet hat. Der Beendigungsschutz verhindert, dass ein Cluster im Falle eines Fehlers heruntergefahren wird. Weitere Informationen erhalten Sie unter [Konfiguration eines Amazon EMR-Clusters zum Fortfahren oder Beenden nach der Ausführung des Schritts](#) und [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).
2. Senden Sie einen Schritt an den Cluster. Weitere Informationen finden Sie unter [Arbeit an einen Amazon EMR-Cluster einreichen](#).
3. Wenn die Verarbeitung des Schritts abgeschlossen ist, prüfen Sie die Schrittprotokolldateien auf Fehler. Weitere Informationen finden Sie unter [Schritt 4: Untersuchen Sie die Amazon EMR-Protokolldateien](#). Die schnellste Möglichkeit zum Auffinden dieser Protokolldateien besteht darin, eine Verbindung mit dem Master-Knoten herzustellen und die Protokolldateien hier anzuzeigen. Die Schrittprotokolldateien werden erst angezeigt, wenn der Schritt einige Zeit ausgeführt wird, beendet wird oder ein Fehler auftritt.
4. Wenn der Schritt erfolgreich ohne Fehler abgeschlossen wurde, führen Sie den nächsten Schritt aus. Wenn Fehler vorliegen, ermitteln Sie den Fehler in den Protokolldateien. Wenn in Ihrem Code ein Fehler aufgetreten ist, korrigieren Sie ihn und führen Sie den Schritt erneut aus. Fahren Sie fort, bis alle Schritte ohne Fehler ausgeführt werden.
5. Wenn Sie das Debuggen des Clusters abgeschlossen haben, müssen Sie den Cluster ggf. manuell beenden. Dies ist erforderlich, da der Cluster mit aktiviertem Beendigungsschutz gestartet wurde. Weitere Informationen finden Sie unter [Verwenden Sie den Kündigungsschutz, um Ihre Amazon EMR-Cluster vor einem versehentlichen Herunterfahren zu schützen](#).

Fehlerbehebung bei einem langsamen Amazon EMR-Cluster

In diesem Abschnitt wird die Fehlerbehebung eines Clusters beschrieben, der noch ausgeführt wird, aber viel Zeit benötigt, um Ergebnisse zurückzugeben. Weitere Informationen zu Verfahren, die Sie anwenden können, wenn der Cluster mit einem Fehlercode beendet wurde, finden Sie unter [Fehlerbehebung bei einem Amazon EMR-Cluster, der mit einem Fehlercode ausgefallen ist](#)

Mit Amazon EMR können Sie die Anzahl und Art der Instances im Cluster festlegen. Diese Spezifikationen sind die beste Möglichkeit, die Geschwindigkeit, mit der Ihre Daten verarbeitet werden, zu beeinflussen. Eine Sache, die Sie in Betracht ziehen könnten, ist die erneute Ausführung des Clusters, wobei Sie diesmal EC2 Instances mit größeren Ressourcen oder eine größere Anzahl von Instances im Cluster angeben. Weitere Informationen finden Sie unter [Amazon EMR-Cluster-Hardware und -Netzwerke konfigurieren](#).

In den folgenden Themen wird erklärt, wie Sie alternative Ursachen für einen langsamen Cluster identifizieren.

Themen

- [Schritt 1: Sammeln Sie Daten über das Problem mit dem Amazon EMR-Cluster](#)
- [Schritt 2: Überprüfen Sie die EMR-Cluster-Umgebung](#)
- [Schritt 3: Untersuchen Sie die Protokolldateien für den Amazon EMR-Cluster](#)
- [Schritt 4: Überprüfen Sie den Zustand des Amazon EMR-Clusters und der Instance](#)
- [Schritt 5: Nach gesperrten Gruppen suchen](#)
- [Schritt 6: Überprüfen Sie die Konfigurationseinstellungen für den Amazon EMR-Cluster](#)
- [Schritt 7: Untersuchen Sie die Eingabedaten für den Amazon EMR-Cluster](#)

Schritt 1: Sammeln Sie Daten über das Problem mit dem Amazon EMR-Cluster

Der erste Schritt bei der Fehlerbehebung bei einem Cluster besteht darin, Informationen darüber zu sammeln, was schief gelaufen ist, sowie über den aktuellen Status und die Konfiguration des Clusters. Diese Informationen werden in den folgenden Schritten verwendet, um mögliche Ursachen des Problems zu bestätigen oder auszuschließen.

Definieren des Problems

Eine klare Definition des Problems ist der erste Ausgangspunkt. Einige Fragen, die Sie sich stellen sollten:

- Was habe ich erwartet? Was ist stattdessen passiert?
- Wann ist dieses Problem zum ersten Mal aufgetreten? Wie oft ist es seitdem passiert?
- Hat sich etwas an der Konfiguration oder Ausführung meines Clusters geändert?

Cluster-Details

Die folgenden Clusterdetails sind hilfreich, um Probleme aufzuspüren. Weitere Informationen zum Sammeln dieser Informationen finden Sie unter [Status und Details des Amazon EMR-Clusters anzeigen](#).

- Die Cluster-ID. (Wird auch als Job-Flow-Identifizier bezeichnet.)
- AWS-Region und Availability Zone, in der der Cluster gestartet wurde.
- Status des Clusters, einschließlich Details zur letzten Statusänderung.
- Typ und Anzahl der für die Master-, Core- und Task-Knoten angegebenen EC2 Instanzen.

Schritt 2: Überprüfen Sie die EMR-Cluster-Umgebung

Prüfen Sie in Ihrer Umgebung, ob es Serviceausfälle gibt oder ob Sie ein AWS Servicelimit überschritten haben.

Themen

- [Prüfen auf Service-Ausfälle](#)
- [Prüfen auf Nutzungsgrenzen](#)
- [Prüfen der Amazon-VPC-Subnetzkonfiguration](#)
- [Neustarten des Clusters](#)

Prüfen auf Service-Ausfälle

Amazon EMR verwendet intern mehrere Amazon Web Services. Es betreibt virtuelle Server auf Amazon EC2, speichert Daten und Skripts auf Amazon S3 und meldet Metriken an CloudWatch. Ereignisse, die diese Services stören, sind selten – wenn sie jedoch auftreten, können sie zu Problemen in Amazon EMR führen.

Überprüfen Sie die [Übersicht zum Servicestatus](#), bevor Sie fortfahren. Prüfen Sie in der Region, in der Sie Ihren Cluster gestartet haben, ob es bei einem dieser Services zu Störungen gekommen ist.

Prüfen auf Nutzungsgrenzen

Wenn Sie einen großen Cluster starten, viele Cluster gleichzeitig gestartet haben oder wenn Sie ein Benutzer sind, der einen Cluster AWS-Konto mit anderen Benutzern teilt, ist der Cluster möglicherweise ausgefallen, weil Sie ein AWS Service-Limit überschritten haben.

Amazon EC2 begrenzt die Anzahl der virtuellen Server-Instances, die in einer einzelnen AWS Region ausgeführt werden, auf 20 On-Demand-Instances oder Reserved Instances. Wenn Sie einen Cluster mit mehr als 20 Knoten starten oder einen Cluster starten, der dazu führt, dass die Gesamtzahl der AWS-Konto auf Ihrem Computer aktiven EC2 Instances 20 überschreitet, kann der Cluster nicht alle benötigten EC2 Instances starten und schlägt möglicherweise fehl. In diesem Fall gibt Amazon EMR einen EC2 QUOTA EXCEEDED-Fehler zurück. Sie können beantragen, dass die Anzahl der EC2 Instances, die Sie auf Ihrem Konto ausführen können, AWS erhöht wird, indem Sie einen [Antrag auf Erhöhung des EC2 Amazon-Instance-Limits](#) einreichen.

Eine weitere Sache, die dazu führen kann, dass Sie Ihre Nutzungslimits überschreiten, ist die Verzögerung zwischen der Beendigung eines Clusters und der Freigabe aller seiner Ressourcen. Je nach Konfiguration kann es bis zu 5–20 Minuten dauern, bis ein Cluster vollständig beendet ist und zugewiesene Ressourcen freigibt. Wenn Sie beim Versuch, einen Cluster zu starten, die Fehlermeldung EC2 QUOTA EXCEEDED erhalten, kann es daran liegen, dass Ressourcen eines kürzlich beendeten Clusters noch nicht zur Verfügung stehen. In diesem Fall können Sie entweder [beantragen, dass Ihr EC2 Amazon-Kontingent erhöht wird](#), oder Sie können zwanzig Minuten warten und den Cluster neu starten.

Amazon S3 begrenzt die Anzahl der auf einem Konto erstellten Buckets auf 100. Wenn Ihr Cluster einen neuen Bucket erstellt, der dieses Limit überschreitet, schlägt die Bucket-Erstellung fehl und kann dazu führen, dass der Cluster fehlschlägt.

Prüfen der Amazon-VPC-Subnetzkonfiguration

Wenn Ihr Cluster in einem Amazon VPC-Subnetz gestartet wurde, muss das Subnetz wie unter [Konfiguration von Netzwerken in einer VPC für Amazon EMR](#) beschrieben konfiguriert werden. Überprüfen Sie außerdem, ob das Subnetz, in dem Sie den Cluster starten, über genügend freie elastische IP-Adressen verfügt, um jedem Knoten im Cluster eine zuzuweisen.

Neustarten des Clusters

Die Verlangsamung der Verarbeitung kann von einer vorübergehenden Bedingung herrühren. Überlegen Sie sich, ob Sie den Cluster beenden und neu starten möchten, um zu prüfen, ob sich die Leistung verbessert.

Schritt 3: Untersuchen Sie die Protokolldateien für den Amazon EMR-Cluster

Der nächste Schritt besteht darin, die Protokolldateien zu untersuchen, um einen Fehlercode oder einen anderen Hinweis auf das Problem zu finden, das in Ihrem Cluster aufgetreten ist. Informationen zu den verfügbaren Protokolldateien, wo sie zu finden sind und wie Sie sie anzeigen können, finden Sie unter [Amazon EMR-Protokolldateien anzeigen](#).

Es kann einige Nachforschungen erfordern, um herauszufinden, was passiert ist. Hadoop führt die Arbeit der Aufträge in Aufgabenversuchen auf verschiedenen Knoten im Cluster aus. Amazon EMR kann spekulative Aufgabenversuche initiieren und die anderen Aufgabenversuche beenden, die nicht zuerst abgeschlossen werden. Dadurch werden umfangreiche Aktivitäten generiert, die in den Controller-, Stderr- und Syslog-Protokolldateien protokolliert werden. Darüber hinaus werden mehrere Aufgaben gleichzeitig ausgeführt, aber eine Protokolldatei kann die Ergebnisse nur linear anzeigen.

Überprüfen Sie zunächst die Bootstrap-Aktionsprotokolle auf Fehler oder unerwartete Konfigurationsänderungen beim Start des Clusters. Suchen Sie anschließend in den Schrittprotokollen nach Hadoop-Aufträgen, die als Teil eines fehlerhaften Schritts gestartet wurden. Untersuchen Sie die Hadoop-Auftragsprotokolle, um die fehlgeschlagenen Aufgabenversuche zu identifizieren. Das Protokoll der Aufgabenversuche wird Details darüber enthalten, was zum Fehlschlagen eines Aufgabenversuchs geführt hat.

In den folgenden Abschnitten wird erläutert, wie die verschiedenen Protokolldateien verwendet werden, um Fehler in Ihrem Cluster zu identifizieren.

Die Bootstrap-Aktionsprotokolle überprüfen

Bootstrap-Aktionen führen Skripts auf dem Cluster aus, während dieser gestartet wird. Sie werden häufig verwendet, um zusätzliche Software auf dem Cluster zu installieren oder um Konfigurationseinstellungen gegenüber den Standardwerten zu ändern. Die Überprüfung dieser Protokolle kann Aufschluss über Fehler geben, die bei der Einrichtung des Clusters aufgetreten sind, sowie über Änderungen der Konfigurationseinstellungen, die sich auf die Leistung auswirken könnten.

Die Schrittprotokolle überprüfen

Es gibt vier Arten von Schrittprotokollen.

- **Controller** – Enthält von Amazon EMR (Amazon EMR) generierte Dateien, die auf Fehler zurückzuführen sind, die bei der Ausführung Ihres Schritts aufgetreten sind. Wenn Ihr Schritt

beim Laden fehlschlägt, finden Sie den Stack-Trace in diesem Protokoll. Fehler beim Laden oder Zugreifen auf Ihre Anwendung werden hier häufig beschrieben, ebenso wie Fehler in der fehlenden Mapper-Datei.

- **stderr** – Enthält Fehlermeldungen, die bei der Verarbeitung des Schritts aufgetreten sind. Fehler beim Laden von Anwendungen werden hier häufig beschrieben. Dieses Protokoll enthält manchmal einen Stack-Trace.
- **stdout** – Enthält den Status, der von Ihren ausführbaren Mapper- und Reducer-Dateien generiert wurde. Fehler beim Laden von Anwendungen werden hier häufig beschrieben. Dieses Protokoll enthält manchmal Anwendungsfehlermeldungen.
- **syslog** – Enthält Protokolle von Software, die nicht von Amazon stammt, wie Apache und Hadoop. Streaming-Fehler werden hier häufig beschrieben.

Überprüfen Sie **stderr** auf offensichtliche Fehler. Wenn **stderr** eine kurze Liste von Fehlern anzeigt, wurde der Schritt schnell beendet und es wurde ein Fehler ausgelöst. Dies wird meistens durch einen Fehler in den Mapper- und Reducer-Anwendungen verursacht, die im Cluster ausgeführt werden.

Untersuchen Sie die letzten Zeilen von Controller und Syslog auf Hinweise auf Fehler oder Ausfälle. Folgen Sie allen Hinweisen zu fehlgeschlagenen Aufgaben, insbesondere wenn dort „Auftrag fehlgeschlagen“ steht.

Die Aufgabenversuchsprotokolle überprüfen

Wenn die vorherige Analyse der Schrittprotokolle eine oder mehrere fehlgeschlagene Aufgaben ergeben hat, suchen Sie in den Protokollen der entsprechenden Aufgabenversuche nach detaillierteren Fehlerinformationen.

Die Hadoop-Daemon-Protokolle überprüfen

In seltenen Fällen kann Hadoop selbst ausfallen. Um zu sehen, ob das der Fall ist, müssen Sie sich die Hadoop-Protokolle ansehen. Sie befinden sich auf `/var/log/hadoop/` auf jedem Knoten.

Sie können die JobTracker Protokolle verwenden, um einen fehlgeschlagenen Taskversuch dem Knoten zuzuordnen, auf dem er ausgeführt wurde. Sobald Sie den Knoten kennen, der mit dem Aufgabenversuch verknüpft ist, können Sie den Zustand der EC2 Instanz überprüfen, die diesen Knoten hostet, um festzustellen, ob Probleme wie etwa ein Mangel an CPU oder Arbeitsspeicher aufgetreten sind.

Schritt 4: Überprüfen Sie den Zustand des Amazon EMR-Clusters und der Instance

Ein Amazon EMR-Cluster besteht aus Knoten, die auf EC2 Amazon-Instances ausgeführt werden. Wenn diese Instances viele Ressourcen binden (z. B. CPU oder Speicherplatz), Probleme mit der Netzwerkkonnektivität haben oder beendet werden, leidet die Geschwindigkeit der Cluster-Verarbeitung.

Es gibt bis zu drei Arten von Knoten in einem Cluster:

- Hauptknoten – verwaltet den Cluster. Wenn ein Leistungsproblem auftritt, ist der gesamte Cluster betroffen.
- Core-Knoten – verarbeiten Map- und Reduce-Aufgaben und verwalten das Hadoop Distributed File System (HDFS). Wenn einer dieser Knoten ein Leistungsproblem hat, kann dies sowohl HDFS-Operationen als auch Map- und Reduce-Verarbeitungen verlangsamen. Sie können einem Cluster zusätzliche Core-Knoten hinzufügen, um die Leistung zu verbessern, aber keine Core-Knoten entfernen. Weitere Informationen finden Sie unter [Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters](#).
- Aufgabenknoten – verarbeiten Map- und Reduce-Aufgaben. Dies sind reine Rechenressourcen und speichern keine Daten. Sie können einem Cluster Aufgabenknoten hinzufügen, um die Leistung zu beschleunigen, oder nicht benötigte Aufgabenknoten entfernen. Weitere Informationen finden Sie unter [Manuelles Ändern der Größe eines laufenden Amazon EMR-Clusters](#).

Wenn Sie den Zustand eines Clusters prüfen, sollten Sie sich sowohl die Leistung des Clusters insgesamt als auch die Leistung der einzelnen Instances anschauen. Es gibt mehrere Tools, die Sie verwenden können:

Überprüfen Sie den Zustand des Clusters mit CloudWatch

Jeder Amazon EMR-Cluster meldet Metriken an CloudWatch. Diese Metriken stellen zusammenfassende Leistungsinformationen über den Cluster bereit, wie z. B. Gesamtlast, HDFS-Auslastung, ausgeführte Aufgaben, verbleibende Aufgaben und beschädigte Blöcke. Wenn Sie sich die CloudWatch Metriken ansehen, erhalten Sie einen Überblick darüber, was in Ihrem Cluster vor sich geht, und Sie erhalten einen Einblick in die Ursachen für die Verlangsamung der Verarbeitung. Sie können nicht nur ein vorhandenes Leistungsproblem analysieren, sondern auch Alarme einrichten, die eine Warnung auslösen CloudWatch , wenn ein future Leistungsproblem auftritt.

CloudWatch Weitere Informationen finden Sie unter [Überwachung von Amazon EMR-Metriken mit CloudWatch](#).

Überprüfen von Auftragsstatus und HDFS-Zustand

Verwenden Sie die Option Application user interface (Anwendungsbenutzeroberflächen) auf der Detailseite des Clusters, um Details zur YARN-Anwendung anzuzeigen. Bei bestimmten Anwendungen können Sie weitere Details und Zugriffsprotokolle direkt anzeigen. Dies ist besonders nützlich für Spark-Anwendungen. Weitere Informationen finden Sie unter [Amazon EMR-Anwendungsverlauf anzeigen](#).

Hadoop bietet eine Reihe von Webschnittstellen, mit denen Sie Informationen anzeigen lassen können. Weitere Informationen darüber, wie Sie auf diese Webschnittstellen zugreifen können, finden Sie unter [Anzeigen von auf Amazon-EMR-Clustern gehosteten Webschnittstellen](#).

- JobTracker — liefert Informationen über den Status des Jobs, der vom Cluster verarbeitet wird. Mit dieser Schnittstelle können Sie ermitteln, wann ein Auftrag blockiert ist.
- HDFS NameNode — liefert Informationen über den Prozentsatz der HDFS-Auslastung und den verfügbaren Speicherplatz auf jedem Knoten. Sie können mit dieser Schnittstelle bestimmen, wann HDFS Ressourcen bindet und zusätzliche Kapazität benötigt.
- TaskTracker — liefert Informationen über die Aufgaben des Jobs, die vom Cluster verarbeitet werden. Mit dieser Schnittstelle können Sie ermitteln, wann eine Aufgabe blockiert ist.

Überprüfen Sie den Zustand Ihrer Instance mit Amazon EC2

Eine andere Möglichkeit, nach Informationen über den Status der Instances in Ihrem Cluster zu suchen, ist die Verwendung der EC2 Amazon-Konsole. Da jeder Knoten im Cluster auf einer EC2 Instance ausgeführt wird, können Sie die von Amazon bereitgestellten Tools verwenden EC2 , um ihren Status zu überprüfen. Weitere Informationen finden Sie unter [Cluster-Instances in Amazon anzeigen EC2](#).

Schritt 5: Nach gesperrten Gruppen suchen

Eine Instance-Truppe wird angehalten, wenn beim Versuch, einen Knoten zu starten, zu viele Fehler auftreten. Wenn z. B. neue Knoten während der Durchführung von Bootstrap-Aktionen wiederholt fehlschlagen, wechselt die Instance-Gruppe nach einiger Zeit in den Status SUSPENDED, anstatt fortlaufend zu versuchen, neue Knoten bereitzustellen.

In folgenden Fällen kann ein Knoten fehlschlagen:

- Hadoop oder der Cluster ist irgendwie beschädigt und akzeptiert keinen neuen Knoten im Cluster.
- Eine Bootstrap-Aktion schlägt auf dem neuen Knoten fehl.
- Der Knoten arbeitet nicht ordnungsgemäß und kann nicht mit Hadoop einchecken

Wenn sich eine Instance-Gruppe im Status `SUSPENDED` befindet und der Cluster den Status `WAITING` hat, können Sie einen Cluster-Schritt hinzufügen, um die gewünschte Anzahl von Core- und Aufgabenknoten zurückzusetzen. Durch Hinzufügen des Schritts wird die Verarbeitung des Clusters fortgesetzt und die Instance-Gruppe wieder in den Status `RUNNING` versetzt.

Weitere Informationen zum Zurücksetzen eines Clusters im angehaltenen Zustand finden Sie unter [Suspendierter Zustand](#).

Schritt 6: Überprüfen Sie die Konfigurationseinstellungen für den Amazon EMR-Cluster

Konfigurationseinstellungen legen die Ausführung eines Clusters im Detail fest, z. B. wie häufig eine Aufgabe wiederholt wird und wie viel Arbeitsspeicher zum Sortieren verfügbar ist. Wenn Sie einen Cluster mithilfe von Amazon EMR starten, gibt es zusätzlich zu den standardmäßigen Hadoop-Konfigurationseinstellungen auch Amazon-EMR-spezifische Einstellungen. Die Konfigurationseinstellungen werden im Master-Knoten des Clusters gespeichert. Sie können die Konfigurationseinstellungen überprüfen, um sicherzustellen, dass Ihr Cluster über die benötigten Ressourcen für einen effizienten Betrieb verfügt.

Amazon EMR legt standardmäßige Hadoop-Konfigurationseinstellungen fest, die zum Starten eines Clusters verwendet werden. Die Werte basieren auf dem AMI und dem Instance-Typ, den Sie für den Cluster angeben. Ändern können Sie die Standardwerte der Konfigurationseinstellungen mithilfe einer Bootstrap-Aktion oder indem Sie neue Werte in den Parametern für die Auftragsausführung festlegen. Weitere Informationen finden Sie unter [Erstellen Sie Bootstrap-Aktionen, um zusätzliche Software mit einem Amazon EMR-Cluster zu installieren](#). Um zu bestimmen, ob eine Bootstrap-Aktion die Konfigurationseinstellungen geändert hat, prüfen Sie die Bootstrap-Aktionsprotokolle.

Amazon EMR protokolliert die Hadoop-Einstellungen für die Ausführung aller Aufträge. Die Protokolldaten werden in einer Datei gespeichert, die `job_job-id_conf.xml` unter dem `/mnt/var/log/hadoop/history/` Verzeichnis des Master-Knotens benannt *job-id* ist und dort durch die ID des Auftrags ersetzt wird. Wenn Sie die Protokollarchivierung aktiviert haben, werden diese Daten in dem `logs/date/jobflow-id/jobs` Ordner nach Amazon S3 kopiert, wo sich das Datum *date* befindet, an dem der Job ausgeführt wurde, und der Identifier des Clusters *jobflow-id* ist.

Die folgenden Konfigurationseinstellungen des Hadoop-Auftrags eignen sich besonders für die Untersuchung von Leistungsproblemen. Weitere Informationen zu den Hadoop-Konfigurationseinstellungen und deren Auswirkungen auf das Verhalten von Hadoop finden Sie unter <http://hadoop.apache.org/docs/>.

 Warning

1. Das Festlegen von `dfs.replication` auf 1 auf Clustern mit weniger als vier Knoten kann zu einem HDFS-Datenverlust führen, wenn ein einzelner Knoten ausfällt. Wir empfehlen, für Produktionsworkloads einen Cluster mit mindestens vier Core-Knoten zu verwenden.
2. Amazon EMR erlaubt Clustern nicht, Core-Knoten unter `dfs.replication` zu skalieren. Bei `dfs.replication = 2` z. B. beträgt die Mindestanzahl von Core-Knoten 2.
3. Wenn Sie verwaltete Skalierung oder Auto-Scaling verwenden oder die Größe Ihres Clusters manuell ändern möchten, empfehlen wir Ihnen, `dfs.replication` auf 2 oder höher einzustellen.

Konfigurationseinstellung	Beschreibung
<code>dfs.replication</code>	Die Anzahl der HDFS-Knoten, in die ein einziger Block (z. B. der Festplattenblock) kopiert wird, um eine RAID-ähnliche Umgebung zu erstellen. Bestimmt die Anzahl der HDFS-Knoten, die eine Kopie des Blocks enthalten.
<code>io.sort.mb</code>	Für die Sortierung verfügbarer Gesamtspeicher. Dieser Wert sollte das Zehnfache von "io.sort.factor" sein. Diese Einstellung kann auch für die Berechnung des vom Aufgabenknoten genutzten Gesamtspeichers durch Berechnen von "io.sort.mb" multipliziert mit "mapred.tasktracker.ap.tasks.maximum" verwendet werden.
<code>io.sort.spill.percent</code>	Wird während der Sortierung verwendet. An diesem Punkt beginnt die Verwendung des Datenträgers, da der für die Sortierung zugewiesene Speicherplatz knapp wird.

Konfigurationseinstellung	Beschreibung
<code>mapred.child.java.opts</code>	Als veraltet gekennzeichnet. Verwenden Sie stattdessen <code>"mapred.map.child.java.opts"</code> und <code>"mapred.reduce.child.java.opts"</code> . Die Java-Optionen, die TaskTracker verwendet werden, wenn eine JVM gestartet wird, damit eine Aufgabe darin ausgeführt wird. <code>"-Xmx"</code> ist ein üblicher Parameter zum Festlegen der maximalen Arbeitsspeichergröße.
<code>mapred.map.child.java.opts</code>	Die Java-Optionen, die beim Starten einer JVM TaskTracker verwendet werden, damit eine Map-Aufgabe darin ausgeführt wird. <code>"-Xmx"</code> ist ein üblicher Parameter zum Festlegen der maximalen Heap-Arbeitsspeichergröße.
<code>mapred.map.tasks.speculative.execution</code>	Legt fest, ob Map-Aufgabenversuche derselben Aufgabe parallel gestartet werden können.
<code>mapred.reduce.tasks.speculative.execution</code>	Legt fest, ob Reduce-Aufgabenversuche derselben Aufgabe parallel gestartet werden können.
<code>mapred.map.max.attempts</code>	Die maximale Anzahl an Map-Aufgabenversuchen. Wenn alle fehlschlagen, wird die Map-Aufgabe als fehlgeschlagen markiert.
<code>mapred.reduce.child.java.opts</code>	Die Java-Optionen, die beim Starten einer JVM für eine Reduce-Aufgabe zur Ausführung innerhalb einer JVM TaskTracker verwendet werden. <code>"-Xmx"</code> ist ein üblicher Parameter zum Festlegen der maximalen Heap-Arbeitsspeichergröße.
<code>mapred.reduce.max.attempts</code>	Die maximale Anzahl an Reduce-Aufgabenversuchen. Wenn alle fehlschlagen, wird die Map-Aufgabe als fehlgeschlagen markiert.

Konfigurationseinstellung	Beschreibung
<code>mapred.reduce.slowstart.completed.maps</code>	Die Anzahl an Map-Aufgaben, die abgeschlossen werden, bevor Reduce-Aufgabenversuche durchgeführt werden. Bei zu geringer Wartezeit kann der Fehler „Too many fetch“ in Versuchen ausgelöst werden.
<code>mapred.reuse.jvm.num.tasks</code>	Eine Aufgabe wird innerhalb einer einzelnen JVM ausgeführt. Gibt an, wie viele Aufgaben dieselbe JVM wiederverwenden dürfen.
<code>mapred.tasktracker.map.tasks.maximum</code>	Die maximale Anzahl von Aufgaben, die während des Map-Vorgangs pro Aufgabenknoten parallel ausgeführt werden können.
<code>mapred.tasktracker.reduce.tasks.maximum</code>	Die maximale Anzahl von Aufgaben, die während des Reduce-Vorgangs pro Aufgabenknoten parallel ausgeführt werden können.

Wenn Ihre Cluster-Aufgaben arbeitsspeicherintensiv sind, können Sie die Leistung verbessern, indem Sie weniger Aufgaben pro Core-Knoten verwenden und die Heap-Größe des JobTrackers reduzieren.

Schritt 7: Untersuchen Sie die Eingabedaten für den Amazon EMR-Cluster

Schauen Sie sich Ihre Eingabedaten an. Sind diese gleichmäßig auf Ihre Schlüsselwerte verteilt? Bei einer starken Datenschiefe in Richtung eines oder weniger Schlüsselwerte wird die Verarbeitungslast möglicherweise einer kleinen Anzahl von Knoten zugeordnet, während sich andere Knoten im Leerlauf befinden. Diese ungleichmäßige Verteilung der Arbeit kann zu einer langsameren Verarbeitung führen.

Um einen ungleichmäßigen Datensatz handelt es sich z. B., wenn ein Cluster ausgeführt wird, um Wörter alphabetisch anzuordnen, aber ein Datensatz zur Verfügung steht, dessen Wörter alle nur mit "a" beginnen. Beim Map-Vorgang wird dann der Knoten überfordert, der Werte verarbeitet, die mit "a" anfangen, während diejenigen Knoten nicht beschäftigt sind, die Wörter mit anderen Anfangsbuchstaben verarbeiten.

Behebung häufiger Probleme bei der Verwendung von Amazon EMR mit AWS Lake Formation

Dieser Abschnitt führt Sie durch die Fehlerbehebung bei Probleme, die häufig bei der Verwendung von Amazon EMR mit AWS Lake Formation auftreten.

Der Zugriff auf den Data Lake ist nicht zulässig

Sie müssen sich ausdrücklich für die Datenfilterung auf Amazon-EMR-Clustern entscheiden, bevor Sie Daten in Ihrem Data Lake analysieren und verarbeiten können. Wenn der Datenzugriff fehlschlägt, wird in der Ausgabe Ihrer Notebookeinträge eine allgemeine `Access is not allowed`-Meldung angezeigt.

Anweisungen dazu, wie Sie die Datenfilterung in Amazon EMR aktivieren und zulassen können, finden Sie unter [Datenfilterung auf Amazon EMR zulassen](#) im AWS Lake Formation - Entwicklerhandbuch.

Sitzungsablauf

Das Sitzungs-Timeout für EMR Notebooks und Zeppelin wird durch die Einstellung `Maximum CLI/API session duration` der IAM-Rolle zur Lake Formation gesteuert. Der Standardwert für diese Einstellung ist eine Stunde. Wenn ein Sitzungs-Timeout auftritt, wird in der Ausgabe Ihrer Notebook-Einträge die folgende Meldung angezeigt, wenn Sie versuchen, Spark SQL-Befehle auszuführen.

```
Error 401    HTTP ERROR: 401 Problem accessing /sessions/2/statements.  
Reason:    JWT token included in request failed validation.  
Powered by Jetty:// 9.3.24.v20180605  
org.springframework.web.client.HttpClientErrorException: 401 JWT token included in  
request failed validation...
```

Aktualisieren Sie die Seite, um Ihre Sitzung zu validieren. Sie werden aufgefordert, sich erneut über Ihren Identitätsanbieter zu authentifizieren und dann zu dem Notebook zurückgeleitet. Sie können nach der erneuten Authentifizierung weitere Abfragen ausführen.

Keine Berechtigungen für Benutzer in der angeforderten Tabelle

Beim Versuch, auf eine Tabelle zuzugreifen, auf die Sie keinen Zugriff haben, wird in der Ausgabe Ihrer Notebook-Einträge die folgende Ausnahme angezeigt, wenn Sie versuchen, Spark SQL-Befehle auszuführen.


```
org.apache.spark.sql.AnalysisException:
  org.apache.hadoop.hive.ql.metadata.HiveException: Unable to fetch table table.
Resource does not exist or requester is not authorized to access requested
permissions.
(Service: AWSGlue; Status Code: 400; Error Code: AccessDeniedException; Request ID: ...
```

Um auf die Tabelle zuzugreifen, müssen Sie dem Benutzer Zugriff gewähren, indem Sie die mit dieser Tabelle verknüpften Berechtigungen in Lake Formation aktualisieren.

Abfragen von kontenübergreifenden Daten, die mit Lake Formation geteilt wurden

Wenn Sie Amazon EMR verwenden, um auf Daten zuzugreifen, die von einem anderen Konto aus mit Ihnen geteilt wurden, versuchen einige Spark-Bibliotheken, den `Glue:GetUserDefinedFunctions`-API-Vorgang aufzurufen. Da die Versionen 1 und 2 der AWS RAM verwalteten Berechtigungen diese Aktion nicht unterstützen, erhalten Sie die folgende Fehlermeldung:

```
"ERROR: User: arn:aws:sts::012345678901:assumed-role/my-
spark-role/i-06ab8c2b59299508a is not authorized to perform:
glue:GetUserDefinedFunctions on resource: arn:exampleCatalogResource
because no resource-based policy allows the glue:GetUserDefinedFunctions
action"
```

Um diesen Fehler zu beheben, muss der Data Lake-Administrator, der die Ressourcenfreigabe erstellt hat, die AWS RAM verwalteten Berechtigungen aktualisieren, die der Ressourcenfreigabe zugeordnet sind. Version 3 der von AWS RAM verwalteten Berechtigungen ermöglicht es Prinzipalen, die `glue:GetUserDefinedFunctions`-Aktion auszuführen.

Wenn Sie eine neue Ressourcenfreigabe erstellen, wendet Lake Formation standardmäßig die neueste Version der AWS RAM verwalteten Berechtigung an, sodass Sie nichts unternehmen müssen. Um den kontenübergreifenden Datenzugriff für bestehende Ressourcenfreigaben zu ermöglichen, müssen Sie die AWS RAM verwalteten Berechtigungen auf Version 3 aktualisieren.

Die AWS RAM Berechtigungen, die Ressourcen zugewiesen wurden, die mit Ihnen geteilt wurden, finden Sie unter AWS RAM. Die folgenden Berechtigungen sind in Version 3 enthalten:

```
Databases
  AWSRAMPermissionGlueDatabaseReadWriteForCatalog
```

```
AWSRAMPermissionGlueDatabaseReadWrite
```

Tables

```
AWSRAMPermissionGlueTableReadWriteForCatalog  
AWSRAMPermissionGlueTableReadWriteForDatabase
```

AllTables

```
AWSRAMPermissionGlueAllTablesReadWriteForCatalog  
AWSRAMPermissionGlueAllTablesReadWriteForDatabase
```

Um die Version mit AWS RAM verwalteten Berechtigungen vorhandener Ressourcenfreigaben zu aktualisieren

Sie (Data Lake-Administrator) können entweder [AWS RAM verwaltete Berechtigungen auf eine neuere Version aktualisieren](#), indem Sie den Anweisungen im AWS RAM Benutzerhandbuch folgen, oder Sie können alle vorhandenen Berechtigungen für den Ressourcentyp widerrufen und sie erneut gewähren. Wenn Sie Berechtigungen widerrufen, wird die mit dem AWS RAM Ressourcentyp verknüpfte Ressourcenfreigabe AWS RAM gelöscht. Wenn Sie Berechtigungen erneut gewähren, AWS RAM erstellt es neue Ressourcenfreigaben, denen die neueste Version der verwalteten Berechtigungen angehängt wird. AWS RAM

Einfügen in, Erstellen und Ändern von Tabellen

Das Einfügen von Daten in Tabellen in Datenbanken und das Erstellen und Ändern von Datenbanken, die durch Lake Formation Richtlinien geschützt sind, wird nicht unterstützt. Wenn Sie diese Operationen ausführen, wird in der Ausgabe Ihrer Notebook-Einträge die folgende Ausnahme angezeigt, wenn Sie versuchen, Spark SQL-Befehle auszuführen:

```
java.io.IOException:  
com.amazon.ws.emr.hadoop.fs.shaded.com.amazonaws.services.s3.model.AmazonS3Exception:  
    Access Denied (Service: Amazon S3; Status Code: 403; Error Code:  
AccessDenied; Request ID: ...
```

Weitere Informationen finden Sie unter [Einschränkungen der Amazon EMR-Integration mit AWS Lake Formation](#).

Schreiben Sie Anwendungen, die Amazon EMR-Cluster starten und verwalten

Sie können auf die von der Amazon EMR-API bereitgestellten Funktionen zugreifen, indem Sie Wrapper-Funktionen in einer der aufrufen. AWS SDKs AWS SDKs Sie bieten sprachspezifische Funktionen, die die API des Webservices umschließen und die Verbindung zum Webservice vereinfachen, indem sie viele Verbindungsdetails für Sie übernehmen. Weitere Informationen zum Aufrufen von Amazon EMR mit einem der finden Sie SDKs unter [Verwenden Sie SDKs , um Amazon EMR anzurufen APIs](#).

Themen

- [End-to-end Beispiel für Amazon EMR Java-Quellcode](#)
- [Allgemeine Konzepte für Amazon EMR-API-Aufrufe](#)
- [Verwenden Sie SDKs , um Amazon EMR anzurufen APIs](#)
- [Amazon EMR Service Quotas verwalten](#)

Important

Die maximale Anforderungsrate für Amazon EMR beträgt eine Anforderung alle zehn Sekunden.

End-to-end Beispiel für Amazon EMR Java-Quellcode

Entwickler können die Amazon-EMR-API über benutzerdefinierten Java-Code aufrufen, um die über die Amazon-EMR-Konsole und CLI verfügbaren Funktionen zu nutzen. Dieser Abschnitt enthält die end-to-end Schritte, die für die Installation AWS Toolkit for Eclipse und Ausführung eines voll funktionsfähigen Java-Quellcode-Beispiels erforderlich sind, das Schritte zu einem Amazon EMR-Cluster hinzufügt.

 Note

Dieses Beispiel konzentriert sich auf Java, aber Amazon EMR unterstützt auch mehrere Programmiersprachen mit einer Sammlung von Amazon SDKs EMR. Weitere Informationen finden Sie unter [Verwenden Sie SDKs , um Amazon EMR anzurufen APIs](#).

In diesem Java-Beispiel wird gezeigt, wie die folgenden Aufgaben mit der Amazon-EMR-API durchgeführt werden:

- AWS Anmeldeinformationen abrufen und an Amazon EMR senden, um API-Aufrufe zu tätigen
- Konfigurieren eines neuen, benutzerdefinierten Schritts und eines neuen, vordefinierten Schritts
- Hinzufügen neuer Schritte zu einem vorhandenen Amazon-EMR-Cluster
- Rufen Sie den Cluster-Schritt IDs aus einem laufenden Cluster ab

 Note

In diesem Beispiel wird gezeigt, wie Sie Schritte zu einem vorhandene, Cluster hinzufügen. Daher ist ein aktiver Cluster in Ihrem Konto erforderlich.

Bevor Sie beginnen, installieren Sie die Version von Eclipse IDE for Java EE Developers, die Ihrer Plattform entspricht. Weitere Informationen erhalten Sie unter [Eclipse-Downloads](#).

Als Nächstes installieren Sie das Database Development Plug-in für Eclipse.

So installieren Sie das Database Development Plug-in für Eclipse

1. Öffnen Sie die Eclipse-IDE.
2. Wählen Sie Help (Hilfe) und dann Install New Software (Neue Software installieren) aus.
3. Geben Sie im Feld Work with: (Arbeiten mit:) **<http://download.eclipse.org/releases/kepler>** oder den Pfad ein, der der Versionsnummer Ihrer Eclipse IDE entspricht.
4. Wählen Sie in der Liste Database Development (Datenbankentwicklung) und Finish (Fertig stellen) aus.
5. Starten Sie Eclipse neu, wenn Sie dazu aufgefordert werden.

Als Nächstes installieren Sie das Toolkit für Eclipse, um hilfreiche, vorkonfigurierte Quellcode-Projektvorlagen nutzen zu können.

So installieren Sie das Toolkit für Eclipse

1. Öffnen Sie die Eclipse-IDE.
2. Wählen Sie Help (Hilfe) und dann Install New Software (Neue Software installieren) aus.
3. Geben Sie im Feld Work with: (Arbeiten mit:) **<https://aws.amazon.com/eclipse>** ein.
4. Wählen Sie in der Artikelliste die Option AWS Toolkit for Eclipse und Fertigstellen aus.
5. Starten Sie Eclipse neu, wenn Sie dazu aufgefordert werden.

Erstellen Sie als Nächstes ein neues AWS Java-Projekt und führen Sie den Java-Beispielquellcode aus.

Um ein neues AWS Java-Projekt zu erstellen

1. Öffnen Sie die Eclipse-IDE.
2. Wählen Sie File (Datei), New (Neu) und Other (Sonstiges) aus.
3. Wählen Sie im Dialogfeld Einen Assistenten auswählen AWS -Java-Projekt und Weiter aus.
4. Geben Sie im Dialogfeld Neues AWS Java-Projekt in das **Project name:** Feld beispielsweise den Namen Ihres neuen Projekts ein **EMR-sample-code**.
5. Wählen Sie AWS Konten konfigurieren..., geben Sie Ihre öffentlichen und privaten Zugangsschlüssel ein und wählen Sie Fertig stellen. Weitere Informationen zum Erstellen von Zugriffsschlüsseln finden Sie unter [Wie erhalte ich Sicherheitsanmeldeinformationen?](#) in Allgemeine Amazon-Web-Services-Referenz.

 Note

Sie sollten Zugriffsschlüssel nicht direkt in den Code einbetten. Das Amazon-EMR-SDK ermöglicht es Ihnen, Zugriffsschlüssel in bekannten Speicherorten abzulegen, sodass Sie sie nicht in den Code integrieren müssen.

6. Klicken Sie im neuen Java-Projekt mit der rechten Maustaste auf den src--Ordner und wählen Sie dann New (Neu) und Class (Klasse) aus.
7. Geben Sie im Dialogfeld Java Class (Java-Klasse) in das Feld Name einen Namen für Ihre neue Klasse ein (z. B. **main**).

8. Wählen Sie im Abschnitt Which method stubs would you like to create? (Welche Method-Stubs möchten Sie erstellen?) `public static void main (String [] args)` und `Finish` (Fertig stellen) aus.
9. Geben Sie den Java-Quellcode in Ihrer neuen Klasse ein und fügen Sie die entsprechenden `import` (Importieren)-Anweisungen für die Klassen und Methoden des Beispiels hinzu. Den vollständigen Quellcode finden Sie unten.

 Note

Ersetzen Sie im folgenden Beispielcode die Beispiel-Cluster-ID (JobFlowId) *j-xxxxxxxxxxxx*, durch eine gültige Cluster-ID in Ihrem Konto, die Sie entweder in AWS Management Console oder finden, indem Sie den folgenden AWS CLI Befehl verwenden:

```
aws emr list-clusters --active | grep "Id"
```

Ersetzen Sie außerdem den Amazon-S3-Beispielpfad *s3://path/to/my/jarfolder* durch den gültigen Pfad der JAR-Datei. Ersetzen Sie den Beispiel-Klassennamen (*com.my.Main1*) durch den richtigen Namen der Klasse in der JAR-Datei (falls relevant).

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

    public static void main(String[] args) {
        AWSCredentials credentials_profile = null;
        try {
            credentials_profile = new
ProfileCredentialsProvider("default").getCredentials();
        } catch (Exception e) {
            throw new AmazonClientException(
```

```

        "Cannot load credentials from .aws/credentials file. " +
        "Make sure that the credentials file exists and the profile name is
specified within it.",
        e);
    }

    AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
        .withCredentials(new AWSStaticCredentialsProvider(credentials_profile))
        .withRegion(Regions.US_WEST_1)
        .build();

    // Run a bash script using a predefined step in the StepFactory helper class
    StepFactory stepFactory = new StepFactory();
    StepConfig runBashScript = new StepConfig()
        .withName("Run a bash script")
        .withHadoopJarStep(stepFactory.newScriptRunnerStep("s3://jeffgoll/emr-scripts/
create_users.sh"))
        .withActionOnFailure("CONTINUE");

    // Run a custom jar file as a step
    HadoopJarStepConfig hadoopConfig1 = new HadoopJarStepConfig()
        .withJar("s3://path/to/my/jarfolder") // replace with the location of the jar
to run as a step
        .withMainClass("com.my.Main1") // optional main class, this can be omitted if
jar above has a manifest
        .withArgs("--verbose"); // optional list of arguments to pass to the jar
    StepConfig myCustomJarStep = new StepConfig("RunHadoopJar", hadoopConfig1);

    AddJobFlowStepsResult result = emr.addJobFlowSteps(new AddJobFlowStepsRequest()
        .withJobFlowId("j-xxxxxxxxxxxx") // replace with cluster id to run the steps
        .withSteps(runBashScript, myCustomJarStep));

    System.out.println(result.getStepIds());

}
}

```

10. Wählen Sie Run (Ausführen), Run As (Ausführen als) und Java Application (Java-Anwendung) aus.
11. Wenn das Beispiel korrekt ausgeführt wird, wird im Fenster der IDs Eclipse IDE-Konsole eine Liste mit den neuen Schritten angezeigt. Die korrekte Ausgabe sieht folgendermaßen oder ähnlich aus:

```
[s-39BLQZRJB2E5E, s-1L6A4ZU2SAURC]
```

Allgemeine Konzepte für Amazon EMR-API-Aufrufe

Wenn Sie eine Anwendung entwickeln, die Amazon-EMR-API-Aufrufe durchführt, gibt es mehrere Konzepte, die Sie beim Aufruf einer der Wrapper-Funktionen in einem SDK einsetzen können.

Themen

- [Endpunkte für Amazon EMR](#)
- [Angaben von Cluster-Parametern in Amazon EMR](#)
- [Availability Zones in Amazon EMR](#)
- [So verwenden Sie weitere Dateien und Bibliotheken in Amazon-EMR-Clustern](#)

Endpunkte für Amazon EMR

Ein Endpunkt ist eine URL, die als Eintrittspunkt für einen Webservice fungiert. Jede Webserviceanforderung muss einen Endpunkt umfassen. Der Endpunkt gibt die AWS Region an, in der Cluster erstellt, beschrieben oder beendet werden. Er hat die Form `elasticmapreduce.regionname.amazonaws.com`. Wenn Sie den allgemeinen Endpunkt (`elasticmapreduce.amazonaws.com`) angeben, leitet Amazon EMR Ihre Anforderung an einen Endpunkt in der Standardregion weiter. Für Konten, die am oder nach dem 8. März 2013 erstellt wurden, lautet die Standardregion "us-west-2"; für ältere Konten ist die Standardregion "us-east-1".

Weitere Informationen über Regionen und Endpunkte für Amazon EMR finden Sie unter [Regionen und Endpunkte](#) im Allgemeine Amazon Web Services-Referenz.

Angaben von Cluster-Parametern in Amazon EMR

Mit den Instances Parametern können Sie den Typ und die Anzahl der EC2 Instanzen konfigurieren, um Knoten zur Verarbeitung der Daten zu erstellen. Hadoop verteilt die Verarbeitung der Daten über mehrere Cluster-Knoten. Der Master-Knoten ist für die Integrität der Core- und Aufgabenknoten sowie für das Abfragen des Auftragsergebnisstatus der Knoten verantwortlich. Die Core- und Aufgabenknoten erledigen die tatsächliche Verarbeitung der Daten. Wenn Sie einen Cluster mit einem Knoten haben, agiert dieser als Master-Knoten und als Core-Knoten.

Der `KeepJobAlive`-Parameter in einer `RunJobFlow`-Anforderung bestimmt, ob der Cluster beendet wird, wenn der Cluster keine auszuführenden Schritte mehr hat. Legen Sie diesen Wert auf `False` fest, wenn Sie wissen, dass der Cluster wie erwartet ausgeführt wird. Bei der Fehlerbehebung des Auftragverlaufs und beim Hinzufügen von Schritten während der ausgesetzten Cluster-Ausführung legen Sie den Wert auf `True` fest. Das reduziert die Zeit und die Kosten für das Hochladen der Ergebnisse in Amazon Simple Storage Service (Amazon S3) der Neustart des Clusters nach dem Bearbeiten eines Schritts müsste wiederholt werden).

`KeepJobAlive` ist dies der `true` Fall, müssen Sie, nachdem Sie den Cluster erfolgreich zum Abschluss gebracht haben, eine `TerminateJobFlows` Anfrage senden. Andernfalls wird der Cluster weiter ausgeführt und es AWS fallen Gebühren an.

Weitere Hinweise zu Parametern, die nur für `getRunJobFlow` gelten, finden Sie unter [RunJobFlow](#). Weitere Informationen zu den grundlegenden Parametern in der Anfrage finden Sie unter [Allgemeine Anforderungsparameter](#).

Availability Zones in Amazon EMR

Amazon EMR verwendet EC2 Instances als Knoten zur Verarbeitung von Clustern. Diese EC2 Instances haben Standorte, die sich aus Availability Zones und Regionen zusammensetzen. Regionen sind verteilt und befinden sich in unterschiedlichen geografischen Zonen. Availability Zones sind eigenständige Standorte innerhalb einer Region, die von Ausfällen anderer Availability Zones isoliert sind. Jede Availability Zone bietet eine kostengünstige Netzwerkkonnektivität mit geringer Latenz zu anderen Availability Zones in der gleichen Region. Eine Liste der Regionen und Endpunkte für Amazon EMR finden Sie unter [Regionen und Endpunkte](#) in der Allgemeinen Amazon Web Services-Referenz.

Der `AvailabilityZone`-Parameter gibt den grundlegenden Speicherort des Clusters an. Dieser Parameter ist optional. Wir empfehlen seine Verwendung. Wenn `AvailabilityZone` nicht angegeben ist, wählt Amazon EMR automatisch den besten `AvailabilityZone`-Wert für den Cluster aus. Der Parameter kann z. B. dann nützlich sein, wenn Sie Ihre Instances mit anderen aktiven Instances gemeinsam platzieren möchten und Ihr Cluster Daten aus diesen Instances lesen oder schreiben muss. Weitere Informationen finden Sie im [EC2 Amazon-Benutzerhandbuch](#).

So verwenden Sie weitere Dateien und Bibliotheken in Amazon-EMR-Clustern

Es kann vorkommen, dass Sie weiteren Dateien oder benutzerdefinierte Bibliotheken für Ihre Mapper oder Reducer-Anwendungen verwenden möchten. Sie können beispielsweise eine Bibliothek nutzen, die eine PDF-Datei in eine Textdatei konvertiert.

So speichern Sie eine Datei für den Mapper oder Reducer bei der Verwendung von Hadoop-Streaming zwischen

- Fügen Sie im JAR-args-Feld das folgende Argument hinzu:

```
-cacheFile s3://bucket/path_to_executable#local_path
```

Die Datei (`local_path`) befindet sich im Arbeitsverzeichnis des Mappers. Dieser kann auf die Datei verweisen.

Verwenden Sie SDKs , um Amazon EMR anzurufen APIs

Themen

- [Verwenden von AWS SDK für Java , um einen Amazon EMR-Cluster zu erstellen](#)

Sie AWS SDKs bieten Funktionen, die die API umschließen und sich um viele Verbindungsdetails kümmern, z. B. um die Berechnung von Signaturen, die Bearbeitung von Wiederholungsversuchen von Anfragen und die Fehlerbehandlung. Sie enthalten SDKs auch Beispielcode, Tutorials und andere Ressourcen, die Ihnen den Einstieg in das Schreiben von Anwendungen erleichtern, die aufrufen AWS. Das Aufrufen der Wrapper-Funktionen in einem SDK kann das Schreiben einer AWS Anwendung erheblich vereinfachen.

Weitere Informationen zum Herunterladen und Verwenden von finden Sie SDKs unter [Tools für Amazon Web Services](#). AWS SDKs

Verwenden von AWS SDK für Java , um einen Amazon EMR-Cluster zu erstellen

Das AWS SDK für Java bietet drei Pakete mit Amazon EMR-Funktionalität:

- [com.amazonaws.services.elasticmapreduce](#)
- [com.amazonaws.services.elasticmapreduce.model](#)
- [com.amazonaws.services.elasticmapreduce.util](#)

Weitere Informationen zu diesen Paketen finden Sie in der [AWS SDK für Java -API-Referenz](#).

Das folgende Beispiel zeigt, wie die Programmierung mit Amazon EMR vereinfacht werden SDKs kann. Das folgende Codebeispiel verwendet das StepFactory-Objekt (eine Hilfsklasse zum Erstellen von typischen Amazon-EMR-Schritttypen) zum Erstellen eines interaktiven Hive-Clusters mit aktiviertem Debugging.

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

    public static void main(String[] args) {
        AWSCredentialsProvider profile = null;
        try {
            credentials_profile = new ProfileCredentialsProvider("default"); // specifies any
            // named profile in
            // .aws/credentials as the credentials provider
        } catch (Exception e) {
            throw new AmazonClientException(
                "Cannot load credentials from .aws/credentials file. " +
                "Make sure that the credentials file exists and that the profile name is defined"
                "within it.",
                e);
        }

        // create an EMR client using the credentials and region specified in order to
        // create the cluster
        AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
            .withCredentials(credentials_profile)
            .withRegion(Regions.US_WEST_1)
```

```

        .build();

// create a step to enable debugging in the AWS Management Console
StepFactory stepFactory = new StepFactory();
StepConfig enableddebugging = new StepConfig()
    .withName("Enable debugging")
    .withActionOnFailure("TERMINATE_JOB_FLOW")
    .withHadoopJarStep(stepFactory.newEnableDebuggingStep());

// specify applications to be installed and configured when EMR creates the
// cluster
Application hive = new Application().withName("Hive");
Application spark = new Application().withName("Spark");
Application ganglia = new Application().withName("Ganglia");
Application zeppelin = new Application().withName("Zeppelin");

// create the cluster
RunJobFlowRequest request = new RunJobFlowRequest()
    .withName("MyClusterCreatedFromJava")
    .withReleaseLabel("emr-5.20.0") // specifies the EMR release version label, we
recommend the latest release
    .withSteps(enableddebugging)
    .withApplications(hive, spark, ganglia, zeppelin)
    .withLogUri("s3://path/to/my/emr/logs") // a URI in S3 for log files is required
when debugging is enabled
    .withServiceRole("EMR_DefaultRole") // replace the default with a custom IAM
service role if one is used
    .withJobFlowRole("EMR_EC2_DefaultRole") // replace the default with a custom EMR
role for the EC2 instance
        // profile if one is used
    .withInstances(new JobFlowInstancesConfig()
        .withEc2SubnetId("subnet-12ab34c56")
        .withEc2KeyName("myEc2Key")
        .withInstanceCount(3)
        .withKeepJobFlowAliveWhenNoSteps(true)
        .withMasterInstanceType("m4.large")
        .withSlaveInstanceType("m4.large"));

RunJobFlowResult result = emr.runJobFlow(request);
System.out.println("The cluster ID is " + result.toString());
}

```

```
}
```

Sie müssen mindestens eine Service- und eine Jobflow-Rolle übergeben, die EMR_ bzw. EMR_ _DefaultRole entsprechen. EC2 DefaultRole Sie können dies tun, indem Sie diesen Befehl für dasselbe Konto aufrufen. AWS CLI Überprüfen Sie zuerst, ob die Rollen bereits vorhanden sind:

```
aws iam list-roles | grep EMR
```

Sowohl das Instanzprofil (EMR_EC2_DefaultRole) als auch die Servicerolle (EMR_DefaultRole) werden angezeigt, sofern sie existieren:

```
"RoleName": "EMR_DefaultRole",  
  "Arn": "arn:aws:iam::AccountID:role/EMR_DefaultRole"  
  "RoleName": "EMR_EC2_DefaultRole",  
  "Arn": "arn:aws:iam::AccountID:role/EMR_EC2_DefaultRole"
```

Wenn die Standardrollen nicht vorhanden sind, können Sie sie über den folgenden Befehl erstellen:

```
aws emr create-default-roles
```

Amazon EMR Service Quotas verwalten

Themen

- [Was sind Amazon EMR Service Quotas?](#)
- [Amazon EMR Service Quotas verwalten](#)
- [Wann sollten EMR-Ereignisse eingerichtet werden in CloudWatch](#)

In den Themen in diesem Abschnitt werden EMR-Dienstkontingente (früher als Service Limits bezeichnet) beschrieben, wie sie in der verwaltet werden und wann es von Vorteil ist AWS Management Console, CloudWatch Ereignisse anstelle von Servicekontingenten zu verwenden, um Cluster zu überwachen und Aktionen auszulösen.

Was sind Amazon EMR Service Quotas?

Ihr AWS Konto verfügt über Standard-Servicekontingenten, auch Limits genannt, für jeden AWS Dienst. Für den EMR-Service gibt es zwei Arten von Grenzwerten:

- Ressourcenbeschränkungen — Sie können EMR verwenden, um EC2 Ressourcen zu erstellen. Diese EC2 Ressourcen unterliegen jedoch Servicekontingenten. Die Ressourcenbeschränkungen in dieser Kategorie sind:
 - Die maximale Anzahl der aktiven Cluster, die gleichzeitig ausgeführt werden können.
 - Die maximale Anzahl aktiver Instances pro Instance-Gruppe.
- Einschränkungen bei APIs — Bei der Verwendung von EMR APIs gibt es zwei Arten von Einschränkungen:
 - Burst-Limit – Dies ist die maximale Anzahl von API-Aufrufen, die Sie gleichzeitig tätigen können. Beispielsweise ist die maximale Anzahl von AddInstanceFleet API-Anfragen, die Sie pro Sekunde stellen können, standardmäßig auf 5 calls/second festgelegt. Dies bedeutet, dass das Burst-Limit der AddInstanceFleet API bei 5 Aufrufen/Sekunde liegt oder dass Sie zu einem bestimmten Zeitpunkt maximal 5 AddInstanceFleet API-Aufrufe tätigen können. Nachdem Sie das Burst-Limit verwendet haben, sind Ihre nachfolgenden Aufrufe jedoch durch das Ratenlimit begrenzt.
 - Ratenlimit – Dies ist die Wiederauffüllrate der Burst-Kapazität der API. Beispielsweise ist die Wiederauffüllrate von AddInstanceFleet Aufrufen standardmäßig auf 0,5 calls/second festgelegt. Das bedeutet, dass Sie, nachdem Sie das Burst-Limit erreicht haben, mindestens 2 Sekunden warten müssen ($0,5 \text{ Aufrufe/Sekunde} \times 2 \text{ Sekunden} = 1 \text{ Anruf}$), um den API-Aufruf zu tätigen. Wenn Sie vorher einen Aufruf tätigen, werden Sie vom EMR-Webservice gedrosselt. Zu jedem Zeitpunkt können Sie nur so viele Aufrufe tätigen, wie die Burst-Kapazität ausreicht, ohne dass dies gedrosselt wird. Mit jeder weiteren Sekunde, die Sie warten, erhöht sich Ihre Burst-Kapazität um 0,5 Aufrufe, bis sie das maximale Limit von 5, dem Burst-Limit, erreicht.

Amazon EMR Service Quotas verwalten

Service Quotas ist eine AWS Funktion, mit der Sie Ihre Amazon EMR-Servicekontingente oder -Limits von einem zentralen Ort aus einsehen und verwalten können, indem Sie die AWS Management Console, die API oder die AWS CLI verwenden. Weitere Informationen zum Anzeigen von Quotas und zum Beantragen einer Erhöhung finden Sie unter [AWS -Service Quotas](#) in der Allgemeine Amazon Web Services-Referenz.

Important

Servicekontingente können unabhängig voneinander durchgesetzt werden, und manchmal können sich ihre Auswirkungen überschneiden. Einige sind auf bestimmte Kontingente

beschränkt APIs, und andere Kontingente setzen Beschränkungen auf Kontoebene durch. Insbesondere werden die folgenden Kontingente auf Kontoebene angewendet:

- Die maximale Rate, mit der Ihr Bucket für alle EMR-Operationen aufgefüllt wird
- Die maximale Anzahl von API-Anfragen, die Sie pro Sekunde stellen können

Beachten Sie, dass es möglich ist, dass eines dieser Kontingente auf Kontoebene weiterhin die API-Aufrufen begrenzt, wenn Sie erfolgreich eine bestimmte Kontingenterhöhung beantragen und weiterhin Drosselungen feststellen, z. B. für API-Anfragen. Zur Problembehebung können Sie die Konsole für Dienstkontingente verwenden, die zu <https://console.aws.amazon.com/servicequotas/Hause> verfügbar ist, um die Kontingentlimits zu überprüfen und Erhöhungen zu beantragen. Wenn Sie Erhöhungen beantragen und weiterhin Drosselungen feststellen, wenden Sie sich an den Support. Standardlimits und Beschreibungen für EMR [Service-Kontingente](#) finden Sie auch im Referenzhandbuch mit AWS allgemeinen Referenzen.

Für einige APIs ist die Einrichtung einer CloudWatch Veranstaltung möglicherweise die bessere Option als die Erhöhung der Servicekontingenten. Sie können auch Zeit sparen, indem CloudWatch Sie proaktiv Alarme einrichten und Erhöhungsanforderungen auslösen, bevor Sie das Servicekontingent erreichen. Weitere Details finden Sie unter [Wann sollten EMR-Ereignisse eingerichtet werden in CloudWatch](#).

Wann sollten EMR-Ereignisse eingerichtet werden in CloudWatch

Bei einigen Abfragen APIs, wie z. B., und DescribeCluster DescribeStep ListClusters, kann die Einrichtung eines CloudWatch Ereignisses die Reaktionszeit auf Änderungen verkürzen und Ihre Servicekontingenten freisetzen. Wenn Sie beispielsweise eine Lambda-Funktion so eingerichtet haben, dass sie ausgeführt wird, wenn sich der Status eines Clusters ändert, z. B. wenn ein Schritt abgeschlossen oder ein Cluster beendet wird, können Sie diesen Auslöser verwenden, um die nächste Aktion in Ihrem Workflow zu starten, anstatt auf die nächste Abfrage zu warten. Andernfalls, wenn Sie über dedizierte EC2 Amazon-Instances oder Lambda-Funktionen verfügen, die ständig die EMR-API nach Änderungen abfragen, verschwenden Sie nicht nur Rechenressourcen, sondern könnten auch Ihr Servicekontingent erreichen.

Im Folgenden sind einige Fälle aufgeführt, in denen Sie von einer Umstellung auf eine ereignisgesteuerte Architektur profitieren könnten.

Fall 1: EMR-Abfrage mithilfe von DescribeCluster API-Aufrufen zur Schrittabwicklung

Example EMR mithilfe von DescribeCluster API-Aufrufen zur Schrittabwicklung abfragen

Ein gängiges Muster besteht darin, einen Schritt an einen laufenden Cluster zu senden und Amazon EMR nach dem Status des Schritts abzufragen, in der Regel mit dem DescribeCluster oder DescribeStep APIs. Diese Aufgabe kann auch mit minimaler Verzögerung erledigt werden, indem Sie sich in das Amazon-EMR-Schrittstatusänderungsereignis einklinken.

Dieses Ereignis enthält die folgenden Informationen in seiner Nutzlast.

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Step Status Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:53:09Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "ERROR",
    "actionOnFailure": "CONTINUE",
    "stepId": "s-ZYXWVUTSRQPON",
    "name": "CustomJAR",
    "clusterId": "j-123456789ABCD",
    "state": "FAILED",
    "message": "Step s-ZYXWVUTSRQPON (CustomJAR) in Amazon EMR cluster j-123456789ABCD (Development Cluster) failed at 2016-12-16 20:53 UTC."
  }
}
```

In der Detailmap könnte eine Lambda-Funktion nach „state“, „stepId“ oder „clusterId“ suchen, um relevante Informationen zu finden.

Fall 2: EMR nach verfügbaren Clustern abfragen, um Workflows auszuführen

Example EMR nach verfügbaren Clustern abfragen, um Workflows auszuführen

Ein Muster für Kunden, die mehrere Cluster ausführen, besteht darin, Workflows auf Clustern auszuführen, sobald sie verfügbar sind. Wenn viele Cluster laufen und ein Workflow auf einem

wartenden Cluster ausgeführt werden muss, könnte ein Muster darin bestehen, EMR mithilfe von API-Aufrufen `DescribeCluster` oder `ListClusters` API-Aufrufen nach verfügbaren Clustern abzufragen. Eine weitere Möglichkeit, die Verzögerung zu verringern, wenn Sie wissen, wann ein Cluster für einen Schritt bereit ist, besteht darin, das Amazon-EMR-Cluster-Statusänderungsereignis in zu verarbeiten.

Dieses Ereignis enthält die folgenden Informationen in seiner Nutzlast.

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}\",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "WAITING",
    "message": "Amazon EMR cluster j-123456789ABCD ..."
  }
}
```

Für dieses Ereignis könnte eine Lambda-Funktion eingerichtet werden, um einen wartenden Workflow sofort an einen Cluster zu senden, sobald sich sein Status in `WAITING` ändert.

Fall 3: EMR nach Cluster-Terminierung abfragen

Example EMR nach Cluster-Terminierung abfragen

Kunden, die viele EMR-Cluster betreiben, fragen häufig bei Amazon EMR nach beendeten Clustern ab, sodass keine Arbeit mehr an Amazon EMR gesendet wird. Sie können dieses Muster mit den `ListClusters` API-Aufrufen `DescribeCluster` und oder mithilfe des Amazon EMR Cluster State Change-Ereignisses in implementieren.

Nach der Clusterbeendigung sieht das ausgegebene Ereignis wie im folgenden Beispiel aus.

```
{
  "version": "0",
```

```
"id": "1234abb0-f87e-1234-b7b6-000000123456",
"detail-type": "EMR Cluster State Change",
"source": "aws.emr",
"account": "123456789012",
"time": "2016-12-16T21:00:23Z",
"region": "us-east-1",
"resources": [],
"detail": {
  "severity": "INFO",
  "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user request\"}",
  "name": "Development Cluster",
  "clusterId": "j-123456789ABCD",
  "state": "TERMINATED",
  "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has terminated at 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
}
```

Der Abschnitt „Detail“ der Nutzlast enthält die ClusterID und den Status, auf den reagiert werden kann.

AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen am Amazon EMR Management Guide seit der letzten Version von Amazon EMR beschrieben. Weitere Informationen zu bestimmten veröffentlichten Versionen von Amazon EMR finden Sie unter [Über Amazon EMR-Versionen](#).

Änderung	Beschreibung	Datum der Veröffentlichung
Aktualisierung der verwalteten Richtlinien	Amazon EMR-Updates für AWS verwaltete Richtlinien — Amazon EMR-Updates für AWS verwaltete Richtlinien — Zusätzliche Berechtigungen für Amazon EMRService Policy_v2.	4. März 2025
Erstversion	Erste Veröffentlichung des Amazon EMR Management Guide für die Versionen 4.x und höher von Amazon EMR. Weitere Informationen zu früheren Versionen von Amazon EMR finden Sie im Amazon EMR Developer Guide .	24. Juli 2015

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.